PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/26, 12/24 — A1

(11) International Publication Number: WO 99/27682

(43) International Publication Date: 3 June 1999 (03.06.99)

(21) International Application Number: PCT/US98/24355

(22) International Filing Date: 13 November 1998 (13.11.98)

(30) Priority Data:
08/976,866 24 November 1997 (24.11.97) US

(71) Applicant: CABLETRON SYSTEMS, INC. [US/US]; 35 Industrial Way, Rochester, NH 03867 (US).

(72) Inventors: LEWIS, Lundy; 480 Greenville Road, Mason, NH 03048 (US). SPARGO, Glenn; 6 Wadsworth Drive, Brookline, NH 03033 (US). DATTA, Utpal; 52 Pinecrest Drive, Bedford, NH 03110 (US).

(74) Agent: HENDRICKS, Therese, A.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND APPARATUS FOR SURVEILLANCE IN COMMUNICATIONS NETWORKS

(57) Abstract

Control of network surveillance in communications networks is accomplished by dividing the surveillance task into two sub–tasks. The first sub–task automatically identifies communications within the network which are to be monitored. Such identification is accomplished by the application of a reasoning system to data received from the network. The identification of the data to be monitored is received by the second sub–task along with network topology information. The second sub–task also applies a reasoning system to this data in order to configure probes and switches within the network so that the identified data can be captured.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

METHOD AND APPARATUS FOR SURVEILLANCE IN COMMUNICATIONS NETWORKS

Background of the Invention

Field of the Invention

The present invention is directed to a method and apparatus for providing surveillance capabilities in a communications network, where the surveillance decisions are made automatically by an analysis of data traversing the network.

Description of Related Art

There is a large amount of traffic flowing through today's computer networks, and not all of this traffic is benign. Thus, the owner or supervisor of the network may need to "listen in on" network communications in order to effectively monitor and secure the network. Such monitoring or surveillance can be achieved by connecting a probe to the network in order to monitor data traveling between two or more nodes (e.g., user workstations) on the network.

Currently, the task of surveillance is "knowledge-intensive," in that human operators generally decide when it is advisable to survey, whom to survey, how long to survey, what kind of information to look for, and how to survey (i.e., where to place the network probes). Thus the surveillance task, as currently known, requires considerable intervention on the part of a human operator.

In a system where communications between two nodes is in a form of discrete packets, the network probe can "read" a packet of data in order to discover information such as the source and destination addressees of the packet, or the protocol of the packet. In addition, over time, measurements can be computed such as the average or total amount of traffic of a certain protocol type during a specific week, or a total number of packets sent to or from a node. This information may then be reported to a system administrator in real-time, or may be stored for later analysis.

Clearview Network Window, a software program available from Clear Communications Corporation, of Lincolnshire, Illinois, USA, allegedly provides predictive/proactive maintenance, intelligent root-cause analysis, and proof-of-quality reports. However, the output is designed for network fault management, which is not the same as "tapping" into a communication between nodes in the network. Thus, the Clearview system does not allow monitoring of data transferred between two nodes in the network with regard to content or characteristics.

Livermore National Laboratory, Livermore, California, USA, developed a group of computer programs to protect the U.S. Department of Energy's computers by "sniffing" data packets that travel across a local area network. The United States Navy used one of these programs, known as the "iWatch" program, in order to wiretap on communications of a suspected computer hacker who had been breaking into computer systems at the U.S. Department of Defense and NASA. The iWatch program uses a network probe to read all packets that travel over a network and then "stores" this information in a common data repository. A simple computer program can then be written to read through the stored data, and to display only "interesting" information. What may be "interesting" is determined by the individual preparing the program and is defined in different ways, e.g., "login names that do not belong to the following: {X, Y, Z, . . .}." Whenever an interesting piece of information is found within the stored data, the stored data is rescanned and a specific number of characters on both sides of the "interesting" piece are reported. These interesting characters are then reviewed in order to determine the content of the message and as a guide to future monitoring activity.

While the iWatch program appears to have been successful in catching at least one computer hacker, it has several limitations. Specifically, the decision to perform a surveillance session on a particular communication node was performed by an individual. This requires that knowledge be conveyed to the individual and that individual make a judgment to proceed with the surveillance. Once the decision to perform the surveillance is started, then all of the data which flows through the node is collected. In other words, the data collection step is not selective. All of the data is collected and stored in a large database for later analysis. Thus, the iWatch method is limited by the size of the database used. In order to provide the most flexibility, large storage units must be set aside, increasing the cost and complexity of the iWatch system. Further, the analysis of the collected data is not performed in real-time. Rather, the software program reads through the stored data in order to determine what is "interesting." Thus, there is a lag between the time that the data is collected, and the analysis to determine if there are communications which should be monitored. This can be a disadvantage since, many times, in order to catch a skilled computer hacker, it is necessary to react immediately to the hacker's presence. Finally, once the "interesting" data has been identified in the iWatch system, once again, an individual operator must make the determination as to where the network probe will be placed in the network in order to "tap" the desired communications. The requirements of human

intervention are thus key steps in the iWatch surveillance system which reduces its efficiency and usefulness.

Summary of the Invention

According to the present invention, a method and apparatus are provided for automatically and intelligently determining when and how to monitor network activity for surveillance purposes.

In a specific embodiment, the system utilizes two reasoning agents which in combination carry out the surveillance task. The inputs and outputs of these agents are defined, but there are several ways to construct the agents depending on the reasoning model or paradigm selected.

In one embodiment, a first reasoning agent receives accounting data from the network which includes a list of communications data sent over the network for a specified time period. The list may include an identification of both the source and destination of the data, and may further identify the protocol used and volume of data sent.

The output of the first reasoning agent (which is provided as an input to the second reasoning agent) may include: whom-to-survey, when-to-survey, and a level-of-surveillance. For example, whom-to-survey may be expressed as communications either: a) sent from a given source; b) delivered to a given destination; or c) sent between a given source and destination. When-to-survey may be expressed as a time interval. Level-of-surveillance may take the form of: volume (data units in/out); protocol; and/or content.

Additional inputs to the second reasoning agent include the network topology and locations of network probes. The goal of the second reasoning agent is to determine which network probes to activate and the instructions needed to set parameters on these network probes in order to monitor, filter and provide the communications of interest (as determined by the output of the first reasoning agent).

By separating the tasks performed by the first and second reasoning agents, and constructing each agent to enhance the separate tasks, a more efficient method of surveillance is achieved.

For example, in a preferred embodiment, a rule-based reasoning system is used for the first reasoning agent, and a constraint-based reasoning system is used for the second reasoning agent, as described in greater detail below.

Surveillance decisions are thus made automatically rather than having decisions made by individuals, and the appropriately programmed tasks analyze the data and implement the surveillance. Specifically, the decision points of: 1) whether and whom to tap; 2) what level of tapping; 3) where to activate probes in the network; and 4) an interpretation of what is heard, can all be automatically accomplished.

The surveillance system of the present invention can be configured to act as either an advisor to a network administrator or configured to work in a fully-automated mode in which decisions are made and necessary actions taken without operator intervention.

The method and apparatus may be implemented in either a router-based or switch-based network, or in a hybrid router/switch-based network.

These and other features and benefits of the present invention will be set forth in the following detailed description and drawings which are given by way of example only and are in no way restrictive.

Brief Description of the Figures

Fig. 1 is a schematic diagram of a network and system incorporating the present invention;

Fig. 2 is a flowchart representing an overview of operations performed in the present invention;

Fig. 3 is a block diagram representation of one embodiment of the present invention;

Fig. 4 is a flowchart showing the steps performed in the identification reasoning agent; and

Fig. 5 is a flowchart representing the steps performed in the probe control reasoning agent.

Detailed Description

A first embodiment of the invention will be described for use in a switch-based network. A switch-based network includes a plurality of devices, such as workstations, printers, storage devices, servers, etc., connected to one another through a plurality of switches. The switches are configured so as to direct a message, usually in the form of a data packet, from a source to a destination. For example, in the MMAC-Plus® system available from Cabletron Systems, Inc., Rochester, New Hampshire, U.S.A., the switches may reside in a common chassis or be

distributed amongst more than one chassis. Although a switch-based network is described, one of ordinary skill in the art will understand that the present invention can be applied in other types of networks.

As shown schematically in Fig. 1, a switched network 100 includes a plurality of switches 102 connected to one another, and a plurality of end nodes 104 each connected to one or more of the switches 102. Data between any two end nodes 104 is sent through at least one switch 102. A network management system 106 includes a topology service, coupled to the network 100 so as to determine the topology of the network and to monitor other network functions. Spectrum®, a network management system available from Cabletron Systems, Inc., polls the network 100 on a regular basis in order to determine the status of the switches 102 and other network devices 104 and maintains information about the topology of the network and about the operations of the network devices.

A processing unit or CPU 108 is connected to the network management system 106 to receive information regarding the operation of the network 100. A memory 110 and storage device 112 are connected to the processor 108 to provide temporary and permanent storage, respectively, of information required by the processor 108. In one embodiment, processor 108 may be running VLAN Manager software available from Cabletron Systems, Inc., which enables "virtual" LANs to be established between different groups of users and/or applications. A display unit 114 is connected to the processor 108 so as to display, generally in graphic form, a representation of the network including its topology and functions. Through either keyboard and/or mouse input devices 116a, 116b, connected to the processor 108, and through the interface program of VLAN Manager, a user can perform various analyses of the network, control the configuration of the network, e.g., adding or deleting nodes and/or switches as the network changes, and monitor data transmissions, as discussed below in more detail.

The VLAN Manager is run on a processor capable of supporting at least one of Windows NT 3.51, Solaris 2.4 and 2.5.1, HP/UX 10.01 and 10.10, AIX 4.0, and IRIX 5.3 operating systems. Any one of a number of commercial or proprietary processors may be used. Generally, the CPU platform 108 requires a minimum of sixty-four Megabytes of RAM, 100 Megabytes of swap space and 150 Megabytes of available disk drive space.

If a user wishes to monitor data or communications between, for example, a source node 104_S and a destination node 104_D in the switched network (see Fig. 1), the user may connect a data analyzer or probe 118 to the network to review the "tapped" data. As disclosed in

commonly assigned and co-pending U.S. patent application Serial No. 08/790,473, entitled "Method and Apparatus to Establish a Tap-Point In a Switched Network Using Self-Configuring Switches Having Distributed Configuration Capabilities," by Liessner et al., (hereinafter "Liessner") which is herein incorporated by reference in its entirety, a user can plug the probe 118 into any switch 102 in the network to which the user has convenient access. Alternatively, a tap-point can be established as disclosed in commonly assigned U.S. patent application Serial No. 08/370,158 entitled "Use of Multipoint Connection Services to Establish Call-Tapping Points in a Switched Network," by Dev et al., (hereinafter "Dev") which is also hereby incorporated by reference in its entirety. In either approach, a probe or tap-point can be established which either receives specific transmissions within the network or is configured to receive all data transmitted by the network.

The probe 118 includes a memory 120 and a storage device 122. In the systems referenced above, the probe 118 may be considered just another device in the switched network, similar to the workstations, printers, storage devices, servers, etc. In addition, there may be multiple probes connected to the switch and/or at other points in the network. As shown, the probe 118 communicates with the CPU 108 over interface 119.

As an overview of the operation of the present invention, a flowchart as shown in Fig. 2 will be referenced. In step 200, accounting data (AD) is received by the processor 108. The accounting data consists of a list of communications over the network for some specified time period. The list may consist of source/destination pairs or may consist of further information such as the communications protocol used and volume of communications for each pair. As the accounting data is received, in step 202, the data is analyzed.

In the present invention, at step 204, traffic on the network which merits further attention is identified. This identification is accomplished automatically and in real-time by the application of reasoning paradigms, e.g., rule-based reasoning, case-based reasoning, constraint-based reasoning, fuzzy logic or neural net analysis. Additional discussion of these and other reasoning paradigm's can be found in Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, Prentice Hall, New Jersey, 1995. By application of any one or more of these reasoning approaches, any traffic on the network which is "suspect" or which requires further analysis is automatically identified. The parameters which define "suspect" traffic or transmissions within the network are set within the reasoning system, as discussed below in more detail.

Once network traffic or data to be tapped or monitored is identified in step 204, the network probe or probes, and/or network switch or switches, are configured in order to collect the data identified in step 206. The identification of the probes and/or switches to be used and/or configured is determined from an analysis of the topology of the network in combination with the system being used for setting up a tap which, as above, can be either the Liessner or Dev systems referenced above. The determination as to how to configure the probe and/or switches is also based upon an application of reasoning approaches which were discussed with regard to step 204. Of course, the criteria for determining which switches and probes to use in order to tap into a given connection in the switched network differs from those used in establishing the criteria for identifying the traffic to be monitored in step 204. Once the probe and switches have been configured, in step 208, the identified traffic is "tapped" and stored for analysis. In this manner, the occurrence of network traffic which merits further attention can be automatically identified without the intervention of an operator and thus accomplished in real-time.

As used in this specification, "real-time" is a matter of degree and not a true/false absolute. Real-time in the short term involves reasoning about those tasks that require close to instantaneous action, with minimal time to think about options, plans, strategies, etc. Real-time in the long term involves reasoning about tasks for which there is time to think about options, plans, etc., i.e., tasks for which action is not urgent.

Within the processing unit 108, the functions as disclosed in steps 202 and 204 are accomplished within an Identification Reasoning (IR) agent 300 as shown in Fig. 3. The IR agent 300 can be implemented as a software program operating within the processing unit 108. The operation of configuring network probes and/or network switches in order to tap identified traffic as per step 206 is performed within a Probe Control Reasoning (PCR) agent 302, which is coupled to the IR agent 300. Similar to the IR agent 300, the PCR agent 302 is a software program which operates on the output from the IR agent 300.

As shown in Fig. 3, the IR agent 300 receives accounting data 304 as an input along with information reasoning (IR) parameters 306. The IR parameters 306 are determined by an operator and are the criteria used by the IR agent 300 in order to identify network traffic or data which merits further attention. The IR parameters 306 include, but are not limited to, particular user names, logical source or destination addresses, physical source or destination addresses, traffic volume thresholds which when exceeded may cause further analysis, communications from or to particular nodes in the network, communications between particular nodes (the classic

"wire-tap"), and communications routed through a particular switch or switches in the network. While nodes are being represented in the preferred embodiment, the present invention would also be applicable to monitoring data communication from/to particular sources or destinations no matter the node at which the source or destination is located since the probe can identify a packet by its source or destination address. The accounting data 304 may include, but once again is not limited to, communications over the network for a specified time period. This information may also include source/destination pairs or may consist of further information such as communications protocol and volume of communications for each pair.

The IR agent 300 monitors traffic in real-time or in a database and is triggered by abnormal events. As an example, the IR agent 300 might simply look at all "spikes" or sudden increases in a parameter and review the sources and destinations of the message units that caused the spike. As a further example, when all traffic data for a particular period of time has been downloaded to an accounting database, for example, the IR agent 300 might be programmed to look for instances of links with exceedingly high volume. Those links that exceed a predetermined threshold would then be chosen for further investigation.

The IR agent 300 applies the IR parameters 306 to the accounting data 304 in order to provide a three part output. Output decision data 307 includes information regarding: 1) who to survey; 2) when to survey; and 3) a level of surveillance. The indication of who to survey could include, but is not limited to, all communications delivered from a given source, all communications delivered to a given destination, or all communications between a given source and destination. The level of surveillance may indicate collection of, for example, the volume of communication, expressed in data units in or out; the protocol being used by the particular message; and/or the contents of the communication, i.e., the message.

The PCR agent 302 receives the who, when and level information from the IR agent 300. The PCR agent 302 also receives probe control reasoning parameters 308 and network topology information 310. The PCR agent 302 automatically applies the network topology information and the reasoning parameters in order to determine probe control output information 312 to configure the probes and switches in order to carry out the monitoring of data as per the output from the IR agent 300.

The probe control output information 312 coming from the PCR agent 302 is in a form such that the network management system 106 is able to configure the switches so as to accomplish the tap. Accordingly, the PCR agent 302 would include information regarding, for

example, either the Liessner method and apparatus, or the Dev multipoint connection service, so that commands can be executed. The PCR agent 302 stores the format structures for a multitude of different networks and/or switching protocols. The network topology information 310 would then include an indication as to the type of network so that the PCR agent 302 could format its probe control information 312 accordingly. Further, a universal standard could be established whereby the probe control information 312 is in a standard format which is not specific to any particular vendor's network management platform. Any network management platform which conforms to the standard would receive this standardized probe control information and translate it so that the tapping connections could be established. In this manner, as new network management platforms become available, the PCR agent need not be updated since its output is of a form that any new network management platform (which complies with the standard) can understand.

Operations within the IR agent 300 will now be discussed in more detail with regard to the flowchart shown in Fig. 4. In step 400, the reasoning parameters are programmed into the IR agent 300. In a preferred embodiment, a rule-based reasoning system has been used in the IR agent 300.

In step 402, the accounting data, as described above, is received by the IR agent 300. The reasoning parameters, according to the rule-based reasoning system, are applied to the received accounting data in step 404. In step 406, the who, when and level results, which are the results of the application of the reasoning parameters to the accounting data, are output. As long as accounting data is received in step 402, steps 404 and 406 are executed. Of course, if necessary, step 400 can be executed when the rules of the rule-based reasoning system need to be changed or updated.

A rule-based reasoning system was chosen for the information reasoning agent since it is relatively easier to understand than case-based reasoning, fuzzy logic, neural networks or other reasoning paradigms. Further, and more importantly, since the monitoring of a network can be expensive, a reasoning paradigm that operates in close to real-time and uses minimal CPU cycles is desirable. A one-ply rule-based system satisfies this requirement since it functions in a manner similar to a look-up table. There are, however, disadvantages associated with a rule-based system since it cannot learn and evolve as the usage of the network evolves. This represents a trade-off between thoroughness and speed. Certainly, depending upon the resources

available and desired thoroughness of analysis, other reasoning systems can be used rather than a rule-based system.

The rules which determine how to identify network communications which are to be monitored are established in the IR agent 300. Merely as examples as to how the rules may function, the following scenarios are provided:

Scenario 1: the network in question is proprietary and all of the users and agents send short and to-the-point messages.

Rule for scenario 1: if any packet is more than X bytes long, then the source of the packet is suspect.

Scenario 2: the network is proprietary, and agents always send messages of protocol type Y.

Rule for scenario 2: if any packet is not of type Y, then the source and destination of the packet are suspects.

Scenario 3: the network is proprietary and it is known that server S should never receive any messages, in other words, there should be no attempts to log onto this server S.

Rule for scenario 3: if any packets have a destination S, then the source of the packet is suspect.

The PCR agent 302 is programmed with the reasoning parameters in step 500 as shown in Fig. 5. A constraint-based reasoning system has been chosen in the preferred embodiment for the PCR agent 302. Constraint-based reasoning was chosen because, at this stage of the surveillance task, the required analysis becomes more complex. The constraints imposed on the PCR agent 302 are the who to survey, when to survey, level of surveillance information, and the network topology information 310 which includes the locations of any available probes.

A goal of constraint-based reasoning is to satisfy as many of the constraints as possible. As an example, the level of surveillance might have to be down-graded from actual content to data units in/out in order to satisfy all the other constraints. Alternatively, the who of surveillance might have to be down-graded from source and destination to only source. In general, there will be several ways to satisfy some, but not all, of the constraints.

As an example, one of the controls in the case-based reasoning system may require that given a choice between down-grading the level of surveillance or who to survey, always down grade the who to survey, setting. It should be noted that the who to survey, when to survey and level of surveillance are "soft-constraints." The placement of probes, however, is typically a "hard-constraint" and the network topology is an even harder constraint.

Once the constraint parameters of the PCR agent 302 are established, the network topology data is received in step 502. The PCR agent 302 is constantly updated with the network topology data so that its perception of the network is accurate. As is known, the topology of a network is dynamic and may change over time. The PCR agent 302 must have information about the topology of the network in order to make proper connections when attempting to tap into communications in the network. In step 504, the who, when and level data are received from the IR agent 300. The constraint-based reasoning algorithms are applied to the network topology data and the data received from the IR agent 300 in step 506. The output from the PCR agent 302, i.e., the probe control data 312, is determined and output in step 508.

This probe control data is used to control the configuration of switches and probes in the network so that the desired data can be monitored. Control then returns to step 502, the receipt of the network topology data, and steps 504, 506, 508 are repeated. The network topology data is constantly received so that existing taps are maintained in the event that the topology of the network changes. In other words, if there is a change to the topology which disrupts the tapping of particular network communications, the PCR agent 302 will respond to the topology change so as to maintain the tapping of the data. This may involve rerouting communications to a probe, using a different probe, or reporting that a tap can no longer be maintained because of a change in the topology of the switching system.

The two reasoning agents 300, 302 in combination carry out the surveillance task. The inputs and the outputs of these agents have been determined, but one of ordinary skill in the art can see that there are several ways to construct the reasoning agents depending on the reasoning paradigm utilized. Thus, for a preferred embodiment, a rule-based reasoning system was selected for the IR agent 300 and a constraint-based reasoning system was chosen for the PCR agent 302, however, it is clear that different reasoning systems may be chosen, respectively, for the agents.

Although the present embodiment is disclosed within the operation of a switch-based network, it is clear that the invention also applies to router-based networks and hybrid

router/switch-based networks. Further, as is known, many kinds of network probes are commercially available. No assumptions nor restrictions about vendor-specific probes have been made. An example of a commonly available probe is the Intelligent RMON/RMON2 Enterprise Probe available from Frontier Software Development, Inc., Chelmsford, MA, USA. This Enterprise Probe uses the RMON standard to provide diagnostic operations for complex network configurations.

Having thus described an embodiment of the present invention, various modifications and improvements will occur to those skilled in the art which are intended to be part of this disclosure and within the scope of the invention. Accordingly, the foregoing description is by way of example only and is not intended as limiting.

CLAIMS

1. A method of monitoring data transmitted between at least two nodes in a network, the method comprising steps of:

(a) receiving, in real-time, data transmitted in the network;

(b) analyzing, in real-time, the retrieved data to identify particular data to be monitored;

(c) monitoring, in real-time, the identified particular data in the network; and

(d) storing the monitored particular data in a storage device.

2. The method as recited in claim 1, wherein step (b) comprises a step of: applying a reasoning operation to the received data to identify the particular data.

3. The method as recited in claim 2, wherein the reasoning operation is a rule-based operation.

4. The method as recited in claim 1, wherein the received data comprises identification of a source of the retrieved data and identification of a destination of the retrieved data.

5. The method as recited in claim 4, wherein the received data further comprises: at least one of a protocol and a volume of data associated with the source and destination.

6. The method as recited in claim 4, wherein step (b) comprises steps of:
applying a rule-based operation to the received data; and
identifying at least one node for performing the monitoring of the particular data, a time period for the monitoring, and a level of the monitoring.

7. The method as recited in claim 6, further comprising at least one step of:
monitoring data delivered to the at least one identified node;
monitoring data sent from the at least one identified node; and

monitoring data sent between the at least one identified node and another node in the network.

8. The method as recited in claim 6, wherein the level of monitoring comprises at least one of:

counting a number of data units;

determining a type of protocol used; and

determining a content of the particular data.

9. An apparatus for monitoring data transmitted between at least two nodes in a network, the apparatus comprising:

means for receiving, in real-time, data transmitted in the network;

means, connected to the receiving means, for analyzing, in real-time, the received data and for identifying particular data for monitoring;

means, connected to the analyzing and identifying means, for monitoring the identified particular data in the network; and

means for storing the monitored particular data.

10. The apparatus as recited in claim 9, wherein the analyzing and identifying means comprise:

means for applying a rule-based reasoning operation to the retrieved data to identify the particular data.

11. The apparatus as recited in claim 10, wherein the monitoring means comprise:

means for applying a constraint-based reasoning operation to monitored particular data.

12. The apparatus as recited in claim 10, wherein the received data comprises identification of a source of the retrieved data and identification of a destination of the retrieved data.

13. The apparatus as recited in claim 12, wherein the received data further comprises at least one of a protocol and volume of data associated with the source and destination.

14. The apparatus as recited in claim 11, wherein the means for analyzing determines at least one of:

at least one node in the network to perform the monitoring;

a time period during which the monitoring is to occur; and

a level of the monitoring.

15. The apparatus as recited in claim 9, wherein the means for analyzing determines at least one of:

a specific node whose output data is to be monitored;

a specific node where all data directed to it is to be monitored; and

a specific source node and a specific destination node wherein all data between the specific source and destination nodes is to be monitored.

16. An apparatus for monitoring data communications in a network, the apparatus comprising:

a first reasoning agent, having a first input to receive accounting data from the network and a second input to receive first reasoning parameters, for generating and outputting identification data by applying the first reasoning parameters to the accounting data according to a first reasoning operation; and

a second reasoning agent, having a third input to receive the identification data from the first reasoning agent, a fourth input to receive second reasoning parameters and a fifth input to receive network topology data, for generating and outputting probe control data by applying the second reasoning parameters to the identification data and the network topology data according to a second reasoning operation.

17. The apparatus according to claim 16, wherein the identification data comprises at least one of:

data identifying at least one node in the network to monitor;

data identifying a time period during which monitoring of the at least one identified node is to occur; and

data indicating a level of the monitoring.

18. The apparatus according to claim 16, wherein the probe control data comprises:

network switch configuration data.

19. The apparatus according to claim 1, wherein the first reasoning operation is a rule-based operation and the second reasoning operation is a constraint-based operation.

20. The apparatus according to claim 16, wherein each of the first and second reasoning agents comprises:

a processing unit; and

a memory unit coupled to the processing unit, the memory unit storing a program according to the respective reasoning operation.

21. An apparatus for monitoring data communications in a network, the apparatus comprising:

a first reasoning agent for identifying data communications within the network to be monitored; and

a second reasoning agent, coupled to the first reasoning agent, for configuring at least one switch within the network to achieve the monitoring of the identified data communication.

22. The apparatus as recited in claim 21, wherein:

the first reasoning agent receives accounting data from the network and outputs identification data by applying a first reasoning operation.

23. The apparatus as recited in claim 22, wherein:

the second reasoning agent receives the identification data from the first reasoning agent and outputs control data by applying a second reasoning operation.

24. The apparatus according to claim 23, wherein the identification data comprises at least one of:

data identifying at least one node in the network to monitor;

data identifying a time period during which monitoring of the at least one identified node is to occur; and

data indicating a level of the monitoring.

25. The apparatus according to claim 23, wherein the probe control data comprises:

network switch configuration data.

26. The apparatus according to claim 23, wherein the first reasoning operation is a rule-based operation and the second reasoning operation is a constraint-based operation.

27. The apparatus according to claim 23, wherein each of the first and second reasoning agents comprises:

a processing unit; and

a memory unit coupled to the processing unit, the memory unit storing a program according to the respective reasoning operation.

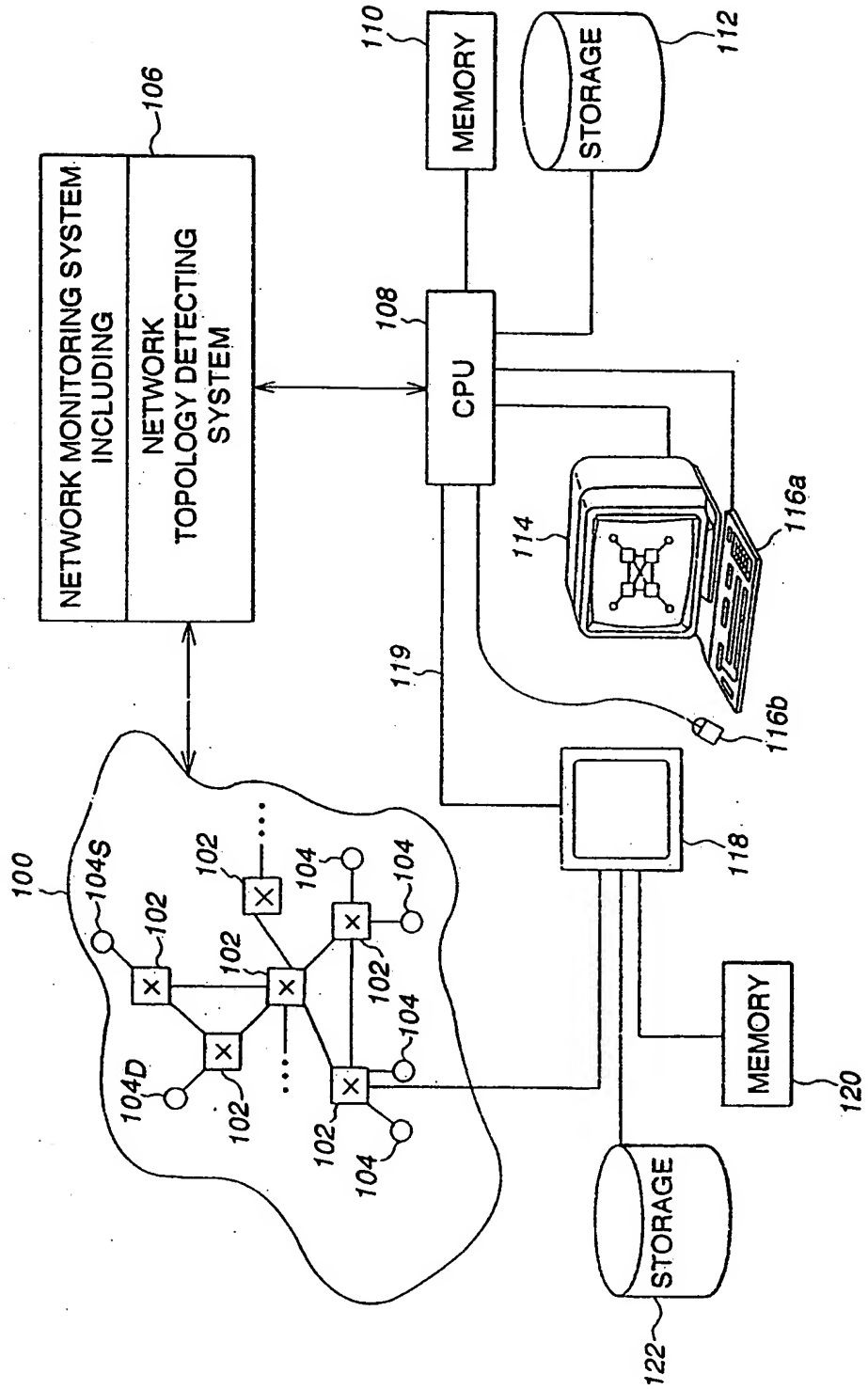

NETWORK MONITORING SYSTEM INCLUDING
NETWORK TOPOLOGY DETECTING SYSTEM
106
MEMORY
110
STORAGE
112
CPU
108
114
116a
116b
119
118
MEMORY
120
STORAGE
122
100
104S
104D
102
104

FIG. 1

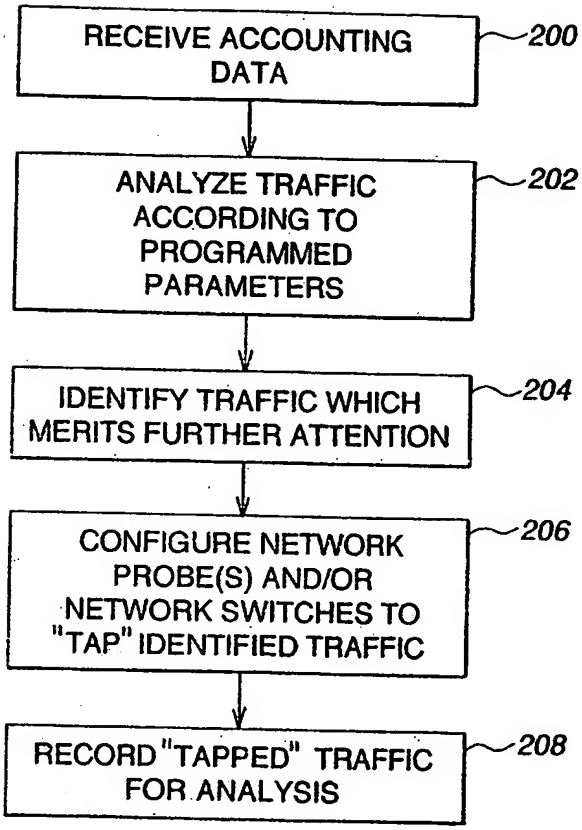

RECEIVE ACCOUNTING DATA
200
ANALYZE TRAFFIC ACCORDING TO PROGRAMMED PARAMETERS
202
IDENTIFY TRAFFIC WHICH MERITS FURTHER ATTENTION
204
CONFIGURE NETWORK PROBE(S) AND/OR NETWORK SWITCHES TO "TAP" IDENTIFIED TRAFFIC
206
RECORD "TAPPED" TRAFFIC FOR ANALYSIS
208

FIG. 2

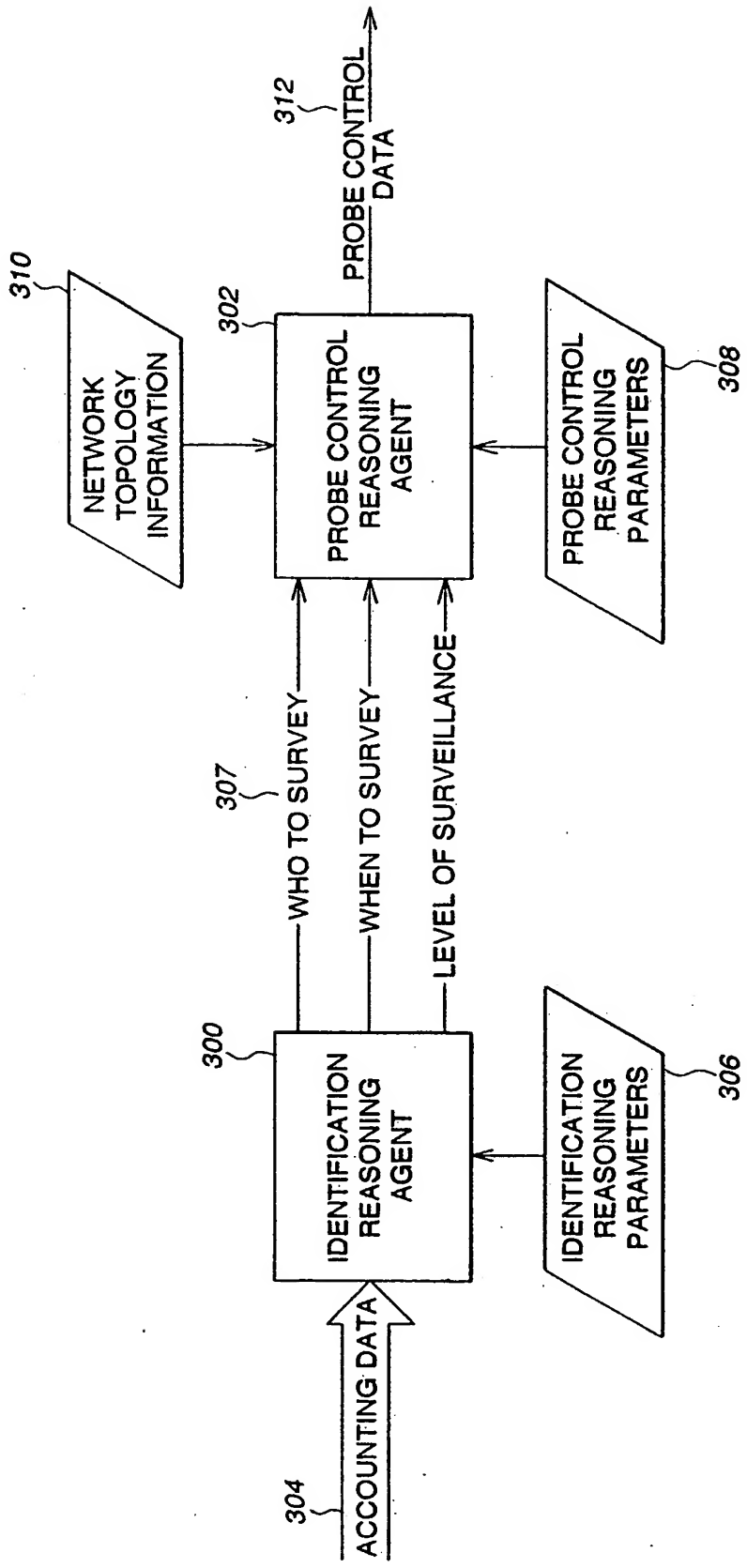

312
PROBE CONTROL DATA
310
NETWORK TOPOLOGY INFORMATION
302
PROBE CONTROL REASONING AGENT
308
PROBE CONTROL REASONING PARAMETERS
307
WHO TO SURVEY
WHEN TO SURVEY
LEVEL OF SURVEILLANCE
300
IDENTIFICATION REASONING AGENT
306
IDENTIFICATION REASONING PARAMETERS
304
ACCOUNTING DATA

FIG. 3

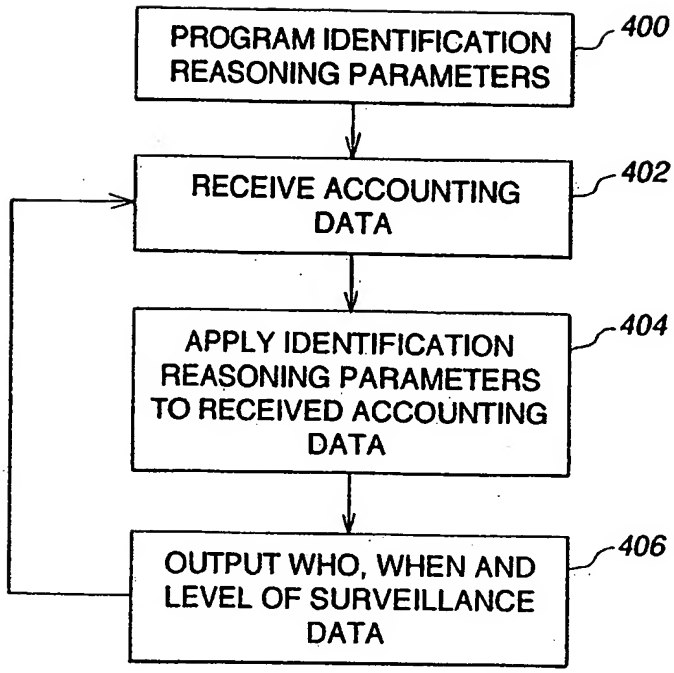

PROGRAM IDENTIFICATION REASONING PARAMETERS
400
RECEIVE ACCOUNTING DATA
402
APPLY IDENTIFICATION REASONING PARAMETERS TO RECEIVED ACCOUNTING DATA
404
OUTPUT WHO, WHEN AND LEVEL OF SURVEILLANCE DATA
406

FIG. 4

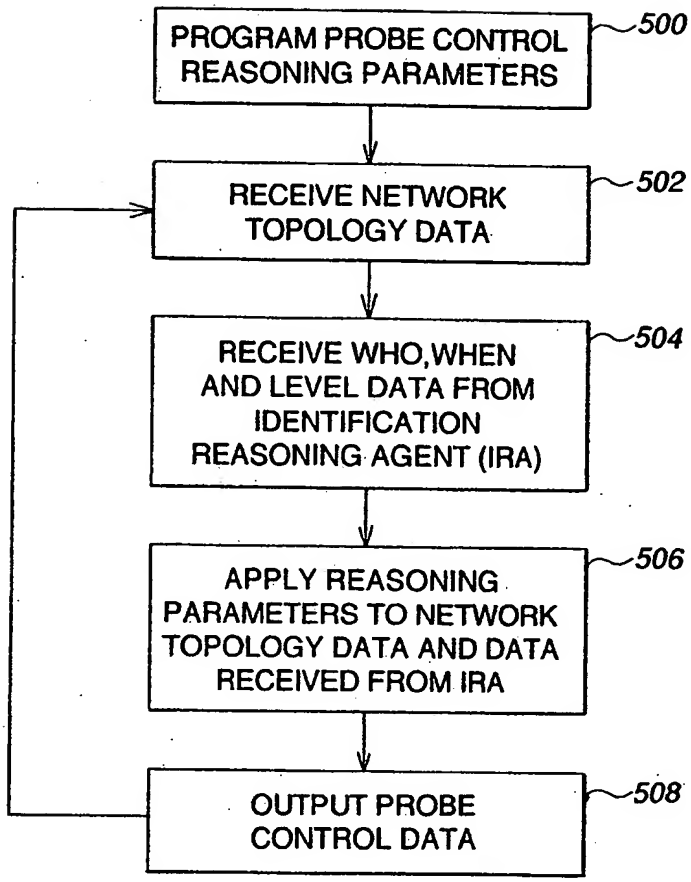

PROGRAM PROBE CONTROL REASONING PARAMETERS
500
RECEIVE NETWORK TOPOLOGY DATA
502
RECEIVE WHO,WHEN AND LEVEL DATA FROM IDENTIFICATION REASONING AGENT (IRA)
504
APPLY REASONING PARAMETERS TO NETWORK TOPOLOGY DATA AND DATA RECEIVED FROM IRA
506
OUTPUT PROBE CONTROL DATA
508

FIG. 5

INTERNATIONAL SEARCH REPORT

Ir ational Application No
PCT/US 98/24355

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L12/26 H04L12/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 478 175 A (HEWLETT PACKARD CO) 1 April 1992 see the whole document ---	1-27
A	"TAILORABLE EMBEDDED EVENT TRACE" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 34, no. 7B, 1 December 1991, pages 259-261, XP000282573 see the whole document ---	1-27
A	LARSEN A K: "ALL EYES ON IP TRAFFIC. NEW APPS CAN MONITOR INTERNET AND INTRANET TRADDIC, BUT DO THEY DELIVER ENOUGH DATA TO HOLD ISPS TO THEIR PROMISES?" DATA COMMUNICATIONS, vol. 26, no. 4, 21 March 1997, pages 54, 56-60, 62, XP000659549 see the whole document ---	1-27

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 March 1999	26/03/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Cichra, M

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

In ational Application No
PCT/US 98/24355

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 634 008 A (GAFFANEY NATHAN J ET AL) 27 May 1997 see the whole document ------	1-27

2

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int ational Application No
PCT/US 98/24355

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0478175 A	01-04-1992	EP 0474932 A	18-03-1992
		DE 69114805 D	04-01-1996
		DE 69114805 T	18-04-1996
		US 5347524 A	13-09-1994
US 5634008 A	27-05-1997	NONE	

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/56 — A1

(11) International Publication Number: WO 99/57858

(43) International Publication Date: 11 November 1999 (11.11.99)

(21) International Application Number: PCT/US99/09853

(22) International Filing Date: 5 May 1999 (05.05.99)

(30) Priority Data:
09/074,059 7 May 1998 (07.05.98) US

(71) Applicant: CABLETRON SYSTEMS, INC. [US/US]; 35 Industrial Way, Rochester, NH 03867 (US).

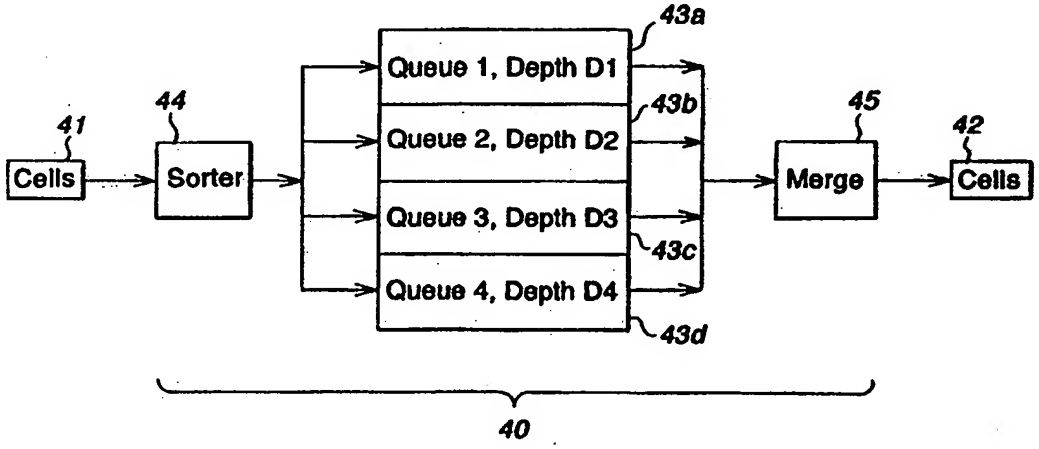
(72) Inventors: DONIS, Marc; 1225 S.W. First Avenue #423, Gainesville, FL 32601 (US). LEWIS, Lundy; 480 Greenville Road, Mason, NH 03048 (US). DATTA, Utpal; 52 Pinecrest Drive, Bedford, NH 03110 (US).

(74) Agent: HENDRICKS, Therese, A.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States: AU, CA, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: MULTIPLE PRIORITY BUFFERING IN A COMPUTER NETWORK



(57) Abstract

Buffer element for communication network, including a first buffer memory to store communication units corresponding to a first quality of service (QOS) level, and a second buffer memory to store communication units corresponding to a second quality of service level. A buffer manager selectively stores communication units from the first and second buffers based on the corresponding quality of service level, and retrieves communication units from the first and second buffer memories. The buffer manager includes a sorter unit for selectively storing based on the quality of service level. The buffer element may further include a depth adjuster to adjust the depth of the first and second buffer memory.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

MULTIPLE PRIORITY BUFFERING IN A COMPUTER NETWORK

Field of the Invention

The invention relates to communication networks and, more particularly, to buffering received and/or transmitted communication units in a communications network.

Discussion of the Related Art

Communication networks have proliferated to enable sharing of resources over a computer network and to enable communications between facilities. A tremendous variety of networks have developed. They may be formed using a variety of different inter-connection elements, such as unshielded twisted pair cables, shield twisted pair cables, shielded cable, fiber optic cable, even wireless inter-connect elements and others. The configuration of these inter-connection elements, and the interfaces for accessing the communication medium, may follow one or more of many topologies (such as star, ring or bus). A variety of different protocols for accessing networking medium have also evolved.

A communication network may include a variety of devices (or "switches") for directing traffic across the network. One form of communication network using switches is an Asynchronous Transfer Mode (ATM) network. These networks route "cells" of communication information across the network. (While the invention may be discussed in the context of ATM networks and cells, this is not intended as limiting.)

FIG. 1 is a block diagram of one embodiment of a network switch 10. In this particular example, the network switch has three input ports 14a-14c and three output ports 14d-14f. The switch is a unidirectional switch, i.e., data flows only in one direction -- from ports 14a-14c to ports 14d-14f. A communication unit (such as an ATM cell, data packet or the like) may be received on one of the ports (e.g., port 14a) and transmitted to any of the output ports (e.g., port 14e). The selection of which output port the communication unit should receive the communication unit may depend on the ultimate destination of the communication unit (and may also depend on the source of the communication unit, in some networks).

Control units 16a-16c route communication units received on the input ports 14a-14c through a switch fabric 12 to the applicable output ports 14d-14f. For example, a communication unit may be received on port 14a. The control unit 16a may route the

communication unit (based, for example, on a destination address contained in the communication unit) through the switch fabric 12 to the buffer 16e. From there, the communication unit is output on port 14e.

The buffers 16d-16f permit the network switch 10 to reconcile varying rates of receiving cells. For example, if a number of cells are received on the various ports 14a -14c, all for the same output port 14d, the output port 14d may not be able to transmit the communication units as quickly as they are received. Accordingly, these units may be buffered.

A great number of variations on the network switch 10 illustrated in FIG. 1 are possible. For example, control unit 16a-16c may be done in a centralized manner. As another example, the buffer in 16d-16f may be done on the input ports (e.g., as part of control units 16a-16c), rather than for the output ports. Another possibility is to use a combined buffer for input and output. This may correspond to pairing an input port with an output port. For example, input port 14a could be paired with output 14d, for the effect of a bi-directional port.

FIG. 2 illustrates buffering using separate receive and transmit buffers at the same time. In this example, network port 24 includes both an input port (e.g., port 25a) and an output port (e.g., 25d). A buffer 26 is provided for the input port. A separate buffer 28 is provided for the output port. Information may be routed through the network switch fabric 22 between ports, as generally described above.

FIG. 3 illustrates an alternative embodiment. In this embodiment, combined receive and transmit buffers are shown. In this embodiment, the receive buffer 36 and transmit buffer are stored in a common memory 35.

Another alternative would be to provide a receive buffer and a transmit buffer that include a shared memory area. Such a system is described in copending and commonly owned United States Patent Application Serial No. 08/847,344, entitled Method And Apparatus For Adaptive Port Buffering, filed April 24, 1997, by Steve Augusta et al., which is hereby incorporated by reference in its entirety.

In many networks, all communication units are treated equally -- i.e., all communication units are assumed to have the same priority in traveling across a network. Alternatively, various levels of quality of service ("QoS") may be provided. This has been applied in ATM networks, although the concept may be applied in other contexts.

In one example, different services offered over the network may have different transmission requirements. For example, video on demand may require high quality service (to avoid jerking movement in the video), while e-mail allows a lower quality of service. Subscribers may be offered the option to pay higher prices for higher levels of quality of service.

Summary of the Invention

According to one embodiment of the present invention, a buffer element for a communication network is disclosed. A first buffer memory is provided to store communication units corresponding to a first quality of service (QoS) level. A second buffer memory stores communication units corresponding to a second quality of service level. A buffer manager is coupled to the first buffer memory and the second buffer memory. A depth adjuster may be provided to adjust corresponding depths of the first buffer memory and the second buffer memory.

According to another embodiment of the present invention, a switch for a communication network is disclosed. The switch includes a plurality of ports, a first buffer memory coupled to one of the ports to store communication units corresponding to a first quality of service level and a second buffer memory coupled to the one of the ports to store communication units corresponding to a second quality of service level.

According to another embodiment of the present invention, a method of buffering communication units in a communication network is disclosed. According to this embodiment, a queue depth is assigned for each of a plurality of queues, each queue being designated to store communication units of a predetermined quality of service level. The plurality of queues is provided, each having the corresponding assigned depth. One of the queues is selected to receive a communication unit, based on a quality of service level associated with the communication unit. The communication unit may then be stored in the selected queue. This embodiment may further comprise a step of adjusting queue depths.

According to another embodiment of the present invention, a method of selecting a communication unit for transmission in a communication network that provides a plurality of quality of service levels is disclosed. In this embodiment, the communication unit is selected from a plurality of communication units stored in a buffer, the buffer including a plurality of queues, each queue corresponding to one of the quality of service levels. The method of this

embodiment includes the steps of identifying the queue with the highest corresponding quality of service level and which is not empty, and then selecting the communication unit from the identified queue.

According to another embodiment of the present invention, a method of storing a communication unit in a buffer is disclosed. According to this embodiment, the communication unit has one of a plurality of quality of service levels and the buffer includes a plurality of queues, each queue corresponding to one of the quality of service levels. According to this embodiment, the method comprises steps of determining the quality of service level of the communication unit and storing the communication unit in the queue having the corresponding quality of service level of the communication unit. According to this embodiment, the communication unit may be dropped when the queue having the corresponding quality of service level of the communication unit is full (or alternatively placed in a queue for a lower quality service).

Brief Description of the Drawings

FIG. 1 illustrates one embodiment of a network switch in a communication network.

FIG. 2 illustrates one embodiment of buffering for a switch.

FIG. 3 illustrates another embodiment of buffering for a switch.

FIG. 4 illustrates one embodiment of a buffer element according to the present invention.

FIG. 5 illustrates one embodiment of a network switch according to the present invention.

FIG. 6 illustrates one embodiment of a method for receiving cells using the buffering element illustrated in FIG. 4.

FIG. 7 illustrates one embodiment of retrieving cells from a buffer element such as that shown in FIG. 4.

FIG. 8 illustrates one embodiment of a method for determining depth assignments for a buffering element.

FIG. 9 illustrates one embodiment of a graphical user interface for inputting queue depth assignment problems.

FIG. 10 illustrates one embodiment of a buffer element and associated controllers for use in a communication network.

FIG. 11 illustrates one embodiment of a method for adjusting queue depths during use of the communication network.

Detailed Description

Design of a communication network (or a switch for use in a communication network) that supports various levels of QoS can be a difficult task. One difficulty is determining the quality of a particular implementation. Generally, the design of a communication network may pursue the following (sometimes conflicting) goals: 1) Accommodating traffic through the network; 2) Making efficient use of the network facilities; 3) Ensuring that network performance reflects the appropriate QoS levels.

Two potential measures of the quality of service offered include cell loss rate (CLR) and cell transfer delay (CTD). CLR reflects the number of cells that are lost. For example, if more cells arrive at a switch than can be accommodated in the switch's buffer, some cells may be lost.

CTD corresponds to the amount of time a cell spends at a switch (or other storage and/or transfer device) before being transmitted. For example, if a cell sits in a buffer for a long period of time while other (e.g., higher QoS level) cells are transmitted, the CTD of the delayed cell is the amount of time it spends in the buffer.

In the embodiment described below, mean cell loss rate (CLR) and mean cell transfer delay (CTD) are used to measure the quality of service. Of course a number of variations on these measures as well as other measures could be used. For example, cell delay variation (the amount of variation in cell delay) or maximum CTD (rather than average CTD) could be used as alternative or additional measures. Other measures may be used instead or as well.

FIG. 4 illustrates one embodiment of a buffer element for use in a network accommodating multiple QoS levels. A buffering mechanism 40 is provided at a switch port, such as the buffering element 16d at port 14d of FIG. 1. In that particular example, the buffering occurs at an output port 14d. In alternative embodiments, buffering may be associated with an input port (e.g., 14a-14c of FIG. 1) or both input and output ports.

In the example of FIG. 4, the buffering element 40 includes four queues (also referred to as buffers) 43a-43d. Each queue is composed of a storage component, such as a random access memory (or any other storage device). Each queue 43a-43d is associated with a particular QoS level for the network. Thus, in the example of FIG. 4, there are four QoS

levels. Queue 1 (43a) corresponds to the highest QoS level. Queue 2 (43b) corresponds to the second highest QoS level. Queue 3 (43c) corresponds to the third highest QoS level. Queue 4 (43d) corresponds to the lowest QoS level.

Each of the queues 43a-43d also has an associated depth. The depth corresponds to the amount of information that can be stored in the particular queue. Where incoming cells 41 have a fixed length, the depth of the queue may be measured by the number of cells that can be stored in that queue.

In Fig. 4, queue 1 (43a) has a depth D1. Queue 2 (43b) has a depth D2. Queue 3 (43c) has a depth D3. Queue 4 (43d) has a depth D4. Each of the depths D1-D4 may be of a different size. When incoming cells 41 are directed to the port, a sorter 44 assigns the cell to the appropriate queue 43a-43d based on the QoS of the cell. In most cases, the QoS of the cell will be indicated in an information field within the cell itself.

When a cell can be transmitted from the port, a merge unit 45 selects the appropriate cell for transmission. While the sorter 44 and merge unit 45 are shown as separate components, these may be implemented in a number of ways. For example, the sorter and merge unit may be separate hardware components. In another embodiment, the sorter 44 and merge unit 45 may be programmed on a general purpose computer coupled to the memory or memories storing queues 43a-43d. In another embodiment, a common merge unit is used for all of the ports (particularly where buffering is done on an input port).

The queues 43a-43d may be implemented using separate memories. In the alternative, the queues may be implemented in a single memory unit, or shared across multiple shared memory units. The memory units may be conventional random access memory device or any other storage element, such as shift registers or other devices.

FIG. 5 illustrates one embodiment of a switch 50 that includes buffering elements 53a, 53b, 54a, 54b, 55a, 55b, 56a and 56b, similar to those illustrated in FIG. 4. The embodiment of FIG. 5 has four input ports 51a-51d and four output ports 52a-52d (and hence is a 4X4 switch).

In the example of FIG. 5, there are only two QoS levels. In this example, each output port 52a-52d has two associated queues (one for each QoS level). For example, output port 52a has two associated queues 53a and 53b. Again, while this embodiment illustrates buffering on the output ports, buffering could instead be done on the input ports or on both

input and output ports. In addition, while FIG. 5 illustrates queues 53a-56b as separate devices, they may be stored in one, or across several, memory chips or other devices.

FIG. 6 illustrates one embodiment of a process for receiving cells at a buffering element, such as receiving incoming cells 41 at buffering element 40 of FIG. 4. The process begins at a step 60 when a cell is received. At a step 61, the appropriate QoS level for the cell is determined. This may be done, for example, by examining a field in the cell that specifies or otherwise indicates the QoS level.

At a step 62, it is determined whether there is room in the appropriate QoS buffer to receive the cell. If so, the cell is stored in the buffer, at a step 63. If there is no room in the appropriate QoS buffer, the cell is dropped at a step 64.

Of course, a number of variations on this process may be developed. As just one example, if there is no room in the appropriate QoS buffer (step 62), buffers of a lower priority could be examined. If there is room in a lower priority buffer, the cell could be stored in that buffer (additional steps may be taken when order of cell transmission is important, such as taking cells from the queue out of FIFO order). In any event, a number of variations and optimization may be made to the embodiment of FIG. 6.

FIG. 7 illustrates one embodiment of a method for retrieving cells stored in a buffering element, such as selecting the outgoing cells 42 of FIG. 4.

In this particular embodiment, the top level queue is selected first (e.g., queue 43a of FIG. 4), at a step 70.

At a step 71, it is determined whether the selected queue is empty. If so, the next queue is selected (at a step 73), and examined to determine if it is empty (step 71).

Once a queue that is not empty has been found, one (or more) cell from that queue is transmitted at a step 72. In this particular embodiment, after a cell has been transmitted, the top level queue is again examined. Accordingly, the effect of the embodiment in FIG. 7 is to transmit cells from the highest level queue that is holding cells, until there are none left.

A number of variations or alternatives are possible. For example, in the embodiment of FIG. 7, a cell in the lowest QoS level queue could be indefinitely frozen from transmission by a long stream of cells arriving for higher level QoS queues. An alternative, therefore, would be to rotate priority among the QoS levels (e.g., give the highest level QoS queue first priority sixty percent of the time, the second highest level priority thirty percent of the time, the third highest level priority ten percent of the time and the lowest QoS level priority none

of the time). Another alternative would be to monitor cell delay and require transmission of cells after a certain delay (the delay potentially depending on the QoS level). For example, queue 3 could be given highest priority when cells have been sitting in that queue for longer than a first period of time, and queue 4 given highest priority when cells have been sitting in that queue for a second period of time (in most cases, the period of time for the lower QoS levels will be greater than the period of time for the higher QoS levels). Again, a number of variations and optimizations are possible.

In the embodiment of FIG. 7, cells are removed from the queue on a first in and first out ("FIFO") basis. Again, a number of alternatives are possible. For example, if a cell is in the highest QoS level queue, but can not be transmitted, another cell may be selected from the highest QoS level queue (or, in the alternative, a cell selected from the next QoS level queue). A cell may not be capable of transmission when, for example, the place to which it is being transmitted is blocked. One example of this situation occurs when the buffers appear at the input ports (e.g., port 14 a of FIG. 1). If another port is transmitting a cell to a particular output port (e.g., port 14d), no other cell stored at any other input port can be transmitted to that same port at the same time. Thus, a cell in the highest QoS level associated with port 14a might be blocked from transmission to port 14d by another cell being transmitted to that port.

Referring again to FIG. 4, the buffering element has M queues, where M stands for the number of levels of QoS accommodated by the switch. In the example of FIG. 4, M equals 4.

Referring again to FIG. 5, an N by N switch is disclosed (in FIG. 5, N=4). Where buffers appear only on the output (or input), there may be a total of $M \times N$ queues in the switch.

In one embodiment of the present invention, each of the queues may have a different depth. That is, the size of each queue may not be the same. In these embodiments, therefore, a problem may be posed of how much memory to provide for each queue, to meet system (and QoS) requirements. This may be referred to as a queue depth assignment problem.

In one embodiment, the assignment of depths to each of the queues is based on performance and characteristic of the network and switch. The depth assignments should satisfy the following equation:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} D_{ij} \leq m$$

Where m is the total memory available in the switch, D_{ij} is the depth of the queue at port i and QoS level is j. Thus, the sum of the depths of all of the queues has to be less than or equal to the total memory (m) available in the switch. As can be seen from this model, the depth of all of the highest quality level queues within the switch may, but need not, be the same. For example, referring again to FIG. 1, more memory could be provided for the highest level queuing associated with port 14d than with port 14e.

One way to determine queue depth is to ascertain a mathematical model for the quality of the queue depth assignments. The mathematical model can then be solved or used to evaluate possible solutions of the depth assignment problem.

In the following example, an energy function is defined to reflect the measure of the quality of the potential solution of the depth assignment problem. In this example, the lower the energy function, the better the solution. The energy function is:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{M} P_{1j} f_1\left(D_{ij}, \mathrm{p}_{ij}\right) \lambda_{ij} + P_{2j} f_2\left(D_{ij}, \mathrm{p}_{ij}, \lambda_{ij}\right),$$

P_{1j} is the constant penalty imposed for a dropped cell on QoS j. (For example, with three QoS levels, weights 10, 5 and 1 could be respectively assigned as the penalty for dropping a cell of the corresponding QoS level.)

P_{2j} is the penalty imposed for a cell waiting on QoS j. (For example, with three QoS levels, penalties of 8, 4 and 0 could be assigned for each unit time delay of a cell having the corresponding QoS level.)

P_{ij} is the load on port i, QoS j, which is given by $\mathrm{p}_{ij} = \lambda_{ij}/\mu_j$. Here, λ_{ij} is the arrival rate, in packets/sec., on port i, QoS j, and μ_j is the processing rate of QoS j, also in packets/sec.

The function $f_1(D, \mathrm{p})$ is the cell loss probability. Therefore, $f_1(D, \mathrm{p})\,\lambda_{ij}$ corresponds to the CLR. The function $f_2(D, \mathrm{p}, \lambda)$ corresponds to the CTD.

To use the above energy function, the particular variables of the equation have to be filled in. Values of λ_{ij} may be determined by observing the traffic over the switch for some

length of time and averaging arrival rates on each queue. Of course, other methods are possible.

The processing rates μ of each queue may be determined by the switch's performance characteristics (or observed).

The penalty parameter arrays P_1 and P_2 may be determined subjectively by the user. These values represent the relative importance of minimizing each of the objective measures f1 and f2 (e.g., CLR and CTD) for each queue. For example, if $P_1 = (10, 5, 2, 0)$, then a penalty of ten is imposed for a lost cell on the first QoS level, a penalty of five on the second QoS level, a penalty of two on the third QoS level, and no penalty on the fourth QoS level. In this example, performance on the fourth QoS level will be sacrificed to improve CLRs of the other QoS levels. Similarly, the penalty associated with cell delay P_2 needs to be specified for each of the QoS levels.

The M/M/1/K queuing model may be used to predict CLR and CTD. This model is discussed, for example in Kleinrock, L., *Queuing Systems, Vol. I: Theory*, New York, NY: John Wiley & Sons, Inc., 1975, pp. 103-5; and Fu, L., *Neural Networks in Computer Intelligence*, New York, NY: McGraw-Hill, Inc., 1994, pp. 41-5. This model assumes that $p < 1$, where p is the load. The cell loss probability, f_1, is given by

$$f_1(D,p) = \frac{(1-p)p^D}{1-p^{D+1}}$$

and the CTD is given by

$$f_2(D,p,\lambda) = \frac{p[1-(D+1)p^D+Dp^{D+1}]}{(1-p^{D+1})(1-p)\lambda(1-f_1(D,p))}$$

(A variety of other models may also be used to predict CLR and CTD. CLR and CTD may also be estimated by taking actual measurements on a system while it is performing.)

One possible approach to solving for minimum E is to examine all possible depth assignments. As is typical of combinatorial problems of this nature, however, the cost of exhaustive search grows factorially. The number of feasible solutions is equal to

$$\frac{(m-1)!}{(m-NM)!(NM-1)!} = \binom{m-1}{NM-1}.$$

Table 1 below illustrates a few examples to show the growth of this function.

Table 1.

m	*NM*	number of possible solutions
30	10	1.00×10^{7}
30	15	7.76×10^{7}
40	10	2.12×10^{8}
40	20	6.89×10^{10}
100	10	1.73×10^{12}
100	25	6.06×10^{22}
100	50	5.04×10^{28}

Under certain embodiments of the present invention, alternative methods may be used to find optimal (or, hopefully, close to optimal) solutions. Thus, neural-networks, genetic algorithms and other approaches may be used.

In one embodiment of the present invention, a straightforward genetic algorithm is used to solve the above energy function. According to this method, an initial solution is started with. This initial solution can be any random solution, or may be selected intelligently as discussed below.

The genetic algorithm then uses a mutation operator that may consist of picking a random port, subtracting a random number from a randomly selected queue on that port and adding that same number to another randomly selected queue depth on the same port. Simple single point cross over may be used to combine solutions. In each generation of the genetic algorithm, an elite percentage of the population is preserved and used to reproduce the remainder of the population using cross over. Half of the offspring may further be mutated a number of times.

In an alternative embodiment, steepest ascent (or descent -- they are the same) hill-climbing (SAHC) may be used. This algorithm (in certain environments) may produce

similar results to that of the genetic algorithm, although in considerably shorter time in certain applications.

Using steepest descent hill-climbing, a local minimum solution can be found by following the steepest path down the energy surface -- following search paths that provide the greatest decreases in the energy function.

The steepest descent hill-climbing approach may be modified to include random jumps. This would permit the algorithm to jump over small "hills" on the energy function surface. This process employs the technique called simulated annealing, known in the art.

The hill-climbing may be achieved by systematically (rather than randomly) incrementing each D_{ij} by one and at the same time reducing the depth of a randomly selected queue by one (thus keeping the total memory usage constant and equal to m). The energy function of each potential solution may be evaluated and the best set of queue depths saved.

For each of the above, an intelligent initial solution can improve the results and/or reduce the amount of time required to achieve a good solution. In one embodiment, the solution is initialized to have queue depths of D_{ij} proportional to p_{ij} $(P_{1j} + P_{2j})$ and summing to exactly m.

Thus, FIG. 8 illustrates one embodiment of a method for finding a solution to the queue depth assignment problem. This embodiment begins at a step 80, where an initial solution is formed. This solution may be formed as described above, assuming that depths D_{ij} are proportional to p_{ij} $(P_{1j} + P_{2j})$ and sum to exactly m.

At a step 88, the current best solution is mutated to determine if a better potential solution may be found. The possible solutions are generated at step 88. For each of the queues at the switch (the queue having an associated depth D_{ij}), the applicable D_{ij} is decreased by one. In addition, a randomly selected queue depth D_{xy} is incremented by one. This forms a new potential solution -- moving one storage element from a current existing queue to a new queue. By both decrementing and adding one, the total memory for the switch remains the same. (Here, the adding and subtracting of one corresponds to adding and subtracting sufficient storage to accommodate one additional cell).

After the new possible solution is generated, its energy function may be evaluated. If this is the best energy function encountered so far, this solution is saved and used for the next iteration (the next time step 88 is performed). Otherwise processing simply continues and the current solution remains the best one encountered so far. Optionally, in the event of a tie, the

newly generated solution is selected. After examining a variety of potential solutions, at step 88, it is determined whether the algorithm has improved the best solution encountered so far at any point in the last (for example) twenty iterations (twenty times passing through step 88). If not the current best solution is taken as the solution to the queue depth problem. If so, the solution has not been stable for the last twenty iterations -- processing continues by returning to step 88 (using the current best solution).

FIG. 9 illustrates one embodiment of a graphical user interface that may be used for solving a queue depth assignment problem. In this particular embodiment, the interface 90 includes an input area 91 and a help area 92. The help area 92 provides a scrollable help document.

As illustrated at 91, the following fields may be input to frame the queue depth assignment problem. A number of switches in the network may be input, as shown at 91a, where more than one switch may be present in the switch fabric.

At 91b, a user may input the number of input and output ports on each switch (N). At 91c, the user may input the number of QoS levels supported by the switch. At 91b, the user may input the total memory available on each switch. (In this embodiment, the input is in terms of the number of cells that can be stored in all of the buffers on the switch.)

At 91e, the user may input the penalty for losing a cell on each QoS level. In the example illustrated in FIG. 9, there are two QoS levels (as shown at 91c). Accordingly, two different entries need to be made at 91e -- one for each QoS level.

Similarly, at 91f, the user inputs the penalties for cell delay on each QoS. As above, the number of entries may correspond to the number of QoS levels (again indicated at 91c).

At 91g, the processing rates (μ) for each quality of service level are input. Finally, at 91h, the arrival rates (λ) for each queue on every switch are input. Thus, in this example, eight entries need to be made -- one for each of the two queues on each of the for output ports.

Tables 2 and 3 below show examples of application of the algorithm of FIG. 8 to the following queue depth assignment problems. Values for λ were determined by two different methods to stimulate mean and maximum load measures. In Table 2, λ values were determined by taking the mean of five random numbers. In Table 3, λ values are the maximum of five random numbers. In both cases, the constraint $\lambda_{ij} < \mu_j$ is enforced.

In all experiments, the number of QoS levels, $M = 4$, $P_1 = (10, 5, 2, 1)$, and $P_2 = (8, 4, 0, 0)$. Values of μ were 100, 60, 30, 15. The Percent Improvement columns show the

improvement over the initial solution (framed using the intelligent solution described above) in each QoS measure for each QoS level. CLRs and CTDs are averaged for each QoS, and are listed in order of QoS level.

Table 2

N	*m*	Final CLR (cells/sec.)	Percent Improvement (%)	Final CTD (sec.)	Percent Improvement (%)	Number of iterations required
4	50	0.460	-278	0.0180	3.75	19
		0.864	110	0.0302	-9.52	
		1.73	141	0.0442	-32.6	
		2.70	-21.2	0.0667	10.0	
4	100	0.0400	-6090	0.0189	0.763	38
		0.741	-7.81	0.0344	0.102	
		0.205	1040	0.0600	-44.1	
		0.374	622	0.118	-76.8	
4	200	0.000538	-6.22×10^6	0.0190	0.0174	87
		0.00109	-79.1	0.0351	0.0208	
		0.00233	36100	0.0659	-27.5	
		0.00653	19000	0.145	-62.9	
6	100	0.154	-722	0.0184	2.00	39
		0.348	32.1	0.0306	-1.20	
		0.910	441	0.0542	-62.7	
		1.39	48.9	0.0827	-12.8	
6	200	0.00838	-70400	0.0188	0.197	82
		0.0184	-53.1	0.0328	0.154	
		0.0414	5920	0.0689	-55.1	
		0.0795	2190	0.129	-66.7	

12	200	0.179 0.313 0.773 1.44	-991 76.6 504 59.2	0.0184 0.0310 0.0544 0.0791	2.41 -3.32 -61.2 -18.7	76
12	500	0.00172 0.00304 0.0104 0.0194	-3.68×10^5 -38.1 10700 9070	0.0190 0.0331 0.0675 0.133	0.0502 0.0238 -30.4 -76.2	94
20	200	0.914 1.76 3.79 2.46	-69.5 49.0 28.8 -2.29	0.0182 0.260 0.0372 0.0667	3.49 -7.28 -11.7 1.43	51
20	500	0.0387 0.0763 0.225 0.415	-3644 26.4 1410 353	0.0200 0.0320 0.0633 0.110	0.798 -0.469 -59.2 -45.5	155
20	1000	0.000572 0.00107 0.00282 0.00663	-4.14×10^5 -160 28100 24700	0.0201 0.0327 0.0695 0.140	0.0204 0.0286 -25.4 -76.0	369

Table 3

N	*m*	Final CLR (cells/sec.)	Percent Improvement (%)	Final CTD (sec.)	Percent Improvement (%)	Number of iterations required
4	50	6.31 7.46 9.28 5.89	-5.14 8.30 0.00 0.00	0.0345 0.0345 0.0333 0.0667	2.69 -4.71 0.00 0.00	7

4	100	2.12 2.74 3.41 5.89	-30.0 5.94 172 0.00	0.0553 0.0561 0.0612 0.0667	7.34 -3.48 -83.5 0.00	20
4	200	0.568 0.772 1.04 2.00	-22.2 3.70 240 128	0.0827 0.0875 0.100 0.148	-0.427 -5.92 967.6 -67.5	46
6	100	4.48 5.20 5.83 6.06	-11.1 9.81 28.1 0.00	0.0424 0.0427 0.0434 0.0667	4.07 -4.40 -14.4 0.00	12
6	200	1.43 1.73 2.34 3.77	-28.3 5.10 187.4 50.1	0.0674 0.0689 0.0711 0.0975	4.12 -2.45 -71.4 -35.4	34
12	200	4.84 5.31 6.17 5.82	-12.1 8.05 36.2 0.00	0.0435 0.0424 0.0435 0.0667	5.92 -2.54 -21.1 0.00	36
12	500	1.07 1.23 1.71 2.70	-23.9 3.01 138 84.9	0.0807 0.0797 0.0867 0.0120	2.74 -2.48 -51.8 -52.0	79
20	200	9.36 11.3 10.0 5.52	-3.27 6.02 0.00 0.00	0.0293 0.0284 0.0333 0.0667	1.78 -3.47 0.00 0.00	14

20	500	2.46 2.98 4.38 5.52	-15.0 6.22 94.1 -3.89	0.0575 0.0595 0.0579 0.0667	3.37 -2.79 -46.7 4.29	57
20	1000	0.731 0.902 1.41 1.94	-27.1 2.74 205 115	0.0870 0.0919 0.108 0.140	2.03 -3.02 -78.9 -58.5	208

As shown in Tables 2 and 3, the new solution is not always superior to the initial solution in all respects. Specifically, the CTD is often worse in the final solution than initially. However, the *overall* goodness of the solution has improved -- some aspects of performance have been sacrificed in order to provide improved measures of aspects deemed more important. In these experiments, CTD was given a comparatively lower priority than CLR, resulting in decreased levels of performance in the CTD measure.

Some of the percentage improvements listed are extremely large in magnitude. These values can be misleading, since the initial quantity may be small. Therefore, even though the percentage is large, the *absolute* change may be of only marginal significance.

A number of problems were also solved by exhaustive search in order to objectively determine optimal solutions for comparison to the SAHC solutions. In every case, the SAHC algorithm found an optimal solution. The problems sizes were necessarily very small, on the order of 10^6 to 10^7. It should be noted, however, that exhaustive search on even these small problems took hours of computation running on a Silicon Graphics Indigo 2 workstation, while the SAHC method was able to arrive at the same solutions in less than one second.

In the above examples, it is assumed that memory could be allocated across all of the buffers in the network. This works well for initial system design.

In an existing system, however, the buffering memories may not be easily reallocated between ports. Referring again to FIG. 1, each of the buffering components 16d-16f are connected to a respective port. After the switch has been designed and built, it may not be convenient to move memory from one of the buffering elements (e.g., 16d) to another buffering element (e.g., 16e). Where this is the case, it may still be possible to optimize queue depths within the individual buffering elements even after the switch has been constructed,

without a shared pool of memory for all buffers on the switch. For example, if each of the queues 43a-43d (of FIG. 4) are stored in a common memory, the amount of memory allocated to each of the buffers may be dynamically changed easily. The technique for assigning queues may be the same as that described above, except that fewer queues are analyzed.

FIG. 10 illustrates one embodiment of a buffering unit according to one embodiment of the present invention, such as the buffering unit 16d of FIG. 1. In this embodiment, a fabric interface controller 102 handles reception of cells from the network switch fabric 100 (in 16d of FIG. 1, this would correspond to reception of cells from the network switch fabric 12). The fabric interface controller may provide cells to the output queue buffers 103 at the direction of a buffer controller 106. Similar to the fabric interface controller 102, a port interface controller 104 handles transmission or reception of cells from the port 105. Both the fabric interface controller 102 and the port interface controller 104 may be implemented as off the shelf devices, or may be integrated into an application specific integrated circuit (ASIC) that includes all or part of the components shown in FIG. 10.

The output queue buffers 103 may be a single dedicated memory device, several memory devices, registers, or a portion of a total memory space used within the switch. As described above, the latter most easily permits assignment and re-aligning of memory among buffering components associated with individual ports, whereas other embodiments may not as easily accommodate this.

In one embodiment, the buffer controller 106 performs the control functions of FIGs. 6-8. This may be done by responding to requests from the fabric interface controller 102 and the port interface controller 104 and controlling the output queue buffers 143 accordingly. In other embodiments, either or both of the fabric interface controller 102 and port interface controller 104 perform some or all of these control functions (as illustrated in FIG. 4), so that a buffer controller 106 is not necessary. In another embodiment, the buffer controller 106 performs the functions of the fabric interface controller 102 and port interface controller 104

The above embodiments also permit dynamic monitoring of network characteristics for the switch or port, and reassignment of queue depths on the fly.

FIG.11 illustrates one embodiment of this process. According to this embodiment, queue depths are assigned at a step 110. This may be done initially as described above, by making assumptions or estimates about network characteristics.

At a step 112, the network characteristics are monitored. These characteristics may

correspond to whatever aspects affect the energy function used in the particular embodiment. For example, in the embodiments described above, mean cell arrival rates (λ), cell drop rates, cell delay rates, average throughput, etc. may be measured. This monitoring may be done by the buffer controller, separate monitoring module, a network controller or other mechanism.

Periodically, the queue depths may be reassigned, by returning to step 110. This may be done at fixed periods of time (e.g., once a day), or may be done whenever a change in network characteristics is sensed. By logging the network characteristics, a schedule of queue depths may be created. This may be useful where the characteristics of the network vary over time (e.g., where network characteristics in the evening are different than network characteristics in the morning).

The process of assigning queue depths 110 may be performed by buffer controllers, as described above with reference to FIG. 10. Even where all of the buffers are held in a common memory and queue depths may be reassigned by sharing memory across more than one port, one or more buffer controllers may be responsible for assigning queue depths. In alternative embodiments, a separate processor may be provided for performing or coordinating the queue depth assignment problem, or this process may be performed by a network controller or other facility.

The various methods above may be implemented as software on a floppy disk, compact disk, or other storage device, which controls a computer. The computer may be a general purpose computer such as a work station, main frame or personal computer, that performs the steps of the disclosed processes or implements equivalents to the disclosed block diagrams. Such a computer typically includes a central processing unit coupled to a random access memory and a program memory by a data bus of some form. The data bus may also be coupled to the output queue. The buffer controller 106 may, for example, perform these functions and be implemented in this manager. Alternatively, the various methods may be implemented in hardware such on an ASIC or other hardware implementation. Of course, in either hardware or software embodiments, functions performed by the above elements and the varying steps may be combined in varying arrangements of hardware and software.

Having thus described at least one illustrative embodiment of the invention, various modifications and improvements will readily occur to those skilled in the art and are intended to be within the scope of the invention. Accordingly, the foregoing description is by way of example only and is not intended as limiting. The invention is limited only as defined in the

following claims and the equivalents thereto.

What is claimed is:

CLAIMS

1. A buffer element for a communication network, the buffer element comprising:

a first buffer memory to store communication units corresponding to a first quality of service level;

a second buffer memory to store communication units corresponding to a second quality of service level; and

a buffer manager, coupled to the first buffer memory and the second buffer memory, to selectively store communication units in the first buffer and the second buffer based on a corresponding quality of service level of the communication units, and to retrieve communication units from the first buffer memory and the second buffer memory.

2. The buffer element of claim 1, wherein the buffer manager comprises:

a sorter unit coupled to the first buffer memory and the second buffer memory to selectively store a communication unit in the first buffer or the second buffer based on a quality of service level of the communication unit.

3. The buffer element of claim 1, wherein the first buffer memory has a first depth, the second buffer memory has a second depth, and the buffer element further comprises:

a depth adjuster to adjust the first depth and the second depth.

4. The buffer element of claim 3, wherein the depth adjuster comprises:

means for iteratively searching possible depth assignments to determine the first depth and the second depth.

5. The buffer element of claim 4, wherein the means for searching comprises:

means for performing a steepest ascent hill climbing search.

6. The buffer element of claim 3, wherein the depth adjuster comprises:

means for determining performance characteristics of the switch.

7. The buffer element of claim 1, wherein the first buffer memory and the second buffer memory are regions of memory in a contiguous random access memory device.

8. The buffer element of claim 1, wherein the communication units are ATM cells.

9. A switch for a communication network, the switch comprising:

a plurality of ports;

a first buffer memory coupled to one of the ports to store communication units corresponding to a first quality of service level; and

a second buffer memory coupled to the one of the ports to store communication units corresponding to a second quality of service level.

10. The switch of claim 9, further comprising:

a buffer manager, coupled to the first buffer memory and the second buffer memory, to selectively store communication units in the first buffer and the second buffer based on a corresponding quality of service level of the communication units, and to retrieve communication units from the first buffer memory and the second buffer memory.

11. The switch of claim 9, wherein:

the plurality of ports comprises a plurality of output ports that output communication units from the switch to the network; and

the first buffer memory and the second buffer memory are coupled to one of the plurality of output ports, to store communication units to be output to the one of the plurality of output ports.

12. The switch of claim 11, wherein:

each of the plurality of output ports has a respective first buffer memory and a respective second buffer memory to store communication units transmitted across the respective output port.

13. The switch of claim 12, wherein:

each of the plurality of output ports has a respective buffer manager to selectively store communication units in the respective first buffer and the respective second buffer based on a corresponding quality of service level of the communication units, and to retrieve communication units from the respective first buffer memory and the respective second buffer memory.

14. The switch of claim 9, wherein:

the plurality of ports comprises a plurality of input ports that receive communication units from the switch to the network; and

the first buffer memory and the second buffer memory are coupled to one of the plurality of input ports, to store communication units received on the one of the plurality of input ports.

15. The switch of claim 14, wherein:

each of the plurality of input ports has a respective first buffer memory and a respective second buffer memory to store communication units transmitted across the respective input port.

16. The switch of claim 15, wherein:

each of the plurality of input ports has a respective buffer manager to selectively store communication units in the respective first buffer and the respective second buffer based on a corresponding quality of service level of the communication unit, and to retrieve communication units from the respective first buffer memory and the respective second buffer memory.

17. The switch of claim 15, wherein the communication units are ATM cells.

18. A method buffering communication units in a communication network, the method comprising steps of:

assigning a queue depth for each of a plurality of queues, each queue being designated to store communication units of a predetermined quality of service level;

providing the plurality of queues, each queue having the corresponding assigned depth;

selecting one of the queues to receive a communication unit based on a quality of service level associated with the communication unit; and

storing the communication unit in the selected queue.

19. The method of claim 18, further comprising a step of adjusting the queue depths.

20. The method of claim 18, further comprising steps of:

monitoring a characteristic in the communication network; and

adjusting the assigned queue depths based on the monitored characteristic.

21. The method of claim 20, wherein the characteristic is selected from the group consisting of communication unit arrival rate for one of the quality of service levels, communication unit processing rate for one of the quality of service levels, communication unit loss rate for one of the quality of service levels and communication unit delay rate for one of the quality of service levels.

22. The method of claim 18, wherein each of the plurality of queues stores communication units for a single port in a communication network switch.

23. The method of claim 22, wherein the single port is an output port.

24. The method of claim 18, wherein the plurality of queues stores the communication units for each port of a switch in the communication network.

25. The method of claim 18, wherein the assigning step comprises a step of:

determining a priority level for dropped communication units for each of the quality of service levels.

26. The method of claim 18, wherein the assigning step comprises a step of:

assigning a priority level for communication unit delay for each of the quality of service levels.

27. The method of claim 18, wherein the assigning step comprises a step of:

performing a search of possible depth assignments.

28. The method of claim 27, wherein the performing step comprises a step of:

performing a steepest ascent hill climbing search.

29. The method of claim 18, wherein the communication units are ATM cells.

30. A method of selecting a communication unit, for transmission in a communication network that provides a plurality of quality of service levels, the communication unit being selected from a plurality of communication units stored in a buffer, the buffer including a plurality of queues, each queue corresponding to one of the quality of service levels, the method comprising steps of:

identifying the queue with the highest corresponding quality of service level and which is not empty; and

selecting the communication unit from the identified queue.

31. A method of storing a communication unit in a buffer, the communication unit having one of a plurality of quality of service levels, the buffer including a plurality of queues, each queue corresponding to one of the quality of service levels, the method comprising steps of:

determining the quality of service level of the communication unit; and

storing the communication unit in the queue having the corresponding quality of service level of the communication unit.

32. The method of claim 31, further comprising a step of:

dropping the communication unit when the queue having the quality of service level of the communication unit is full.

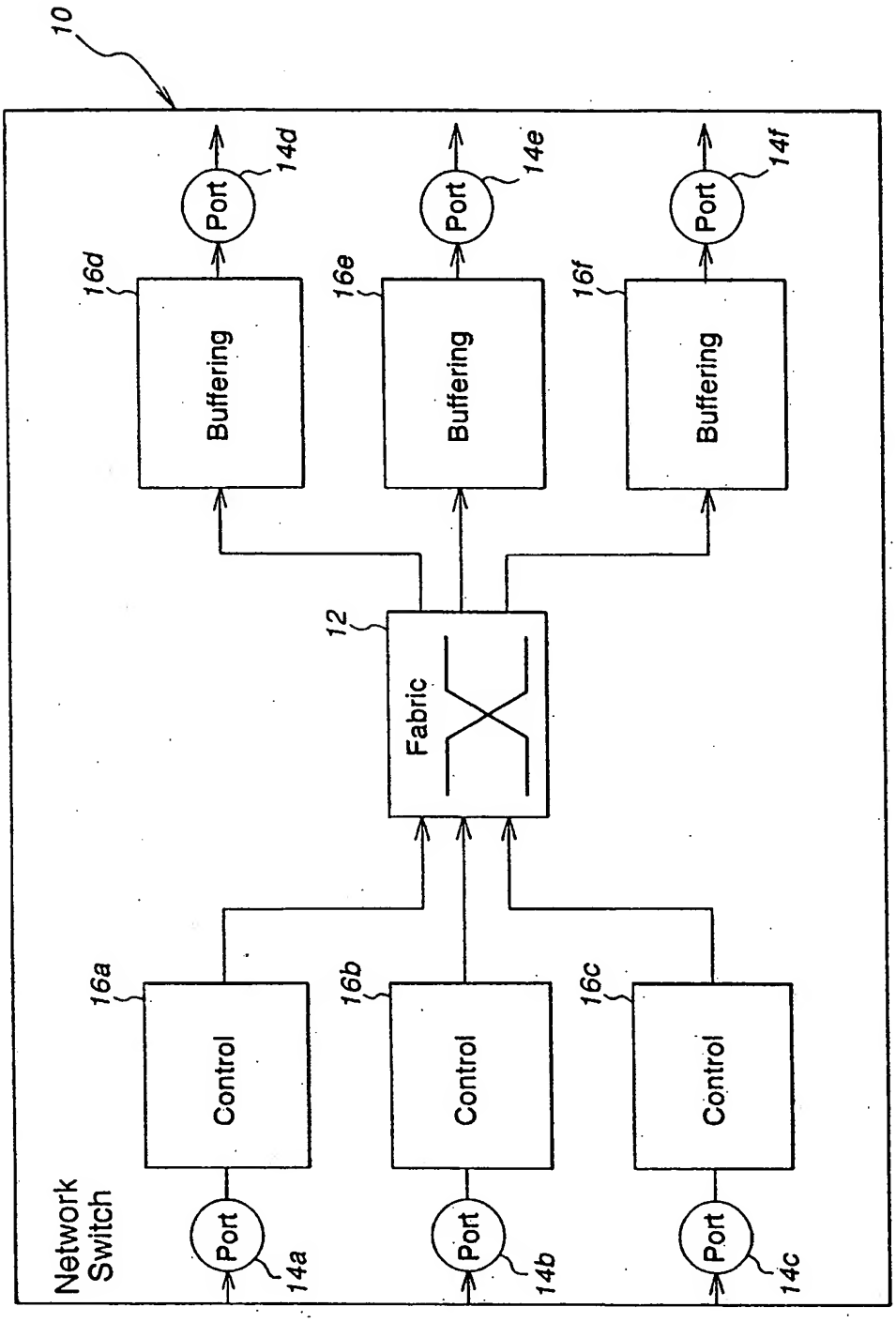


FIG. 1
(PRIOR ART)

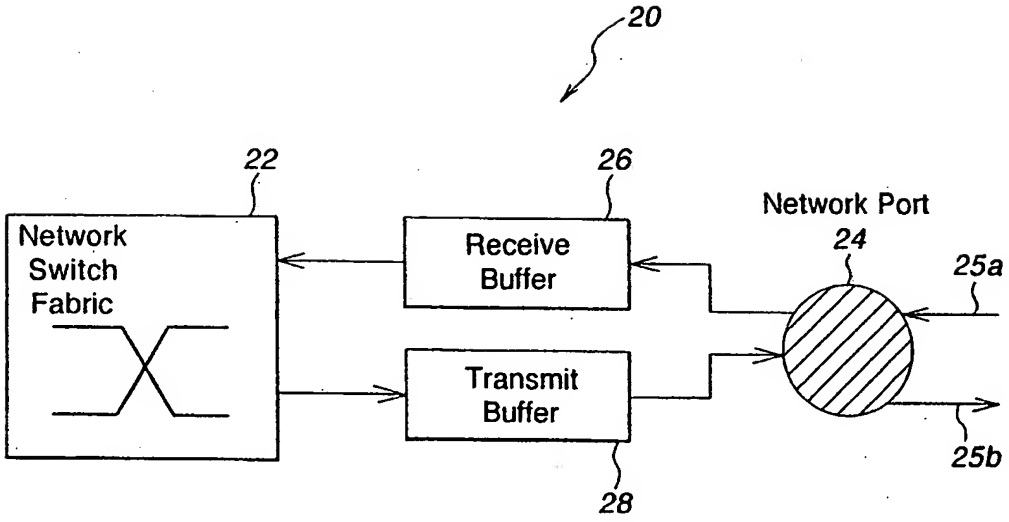


FIG. 2

(PRIOR ART)

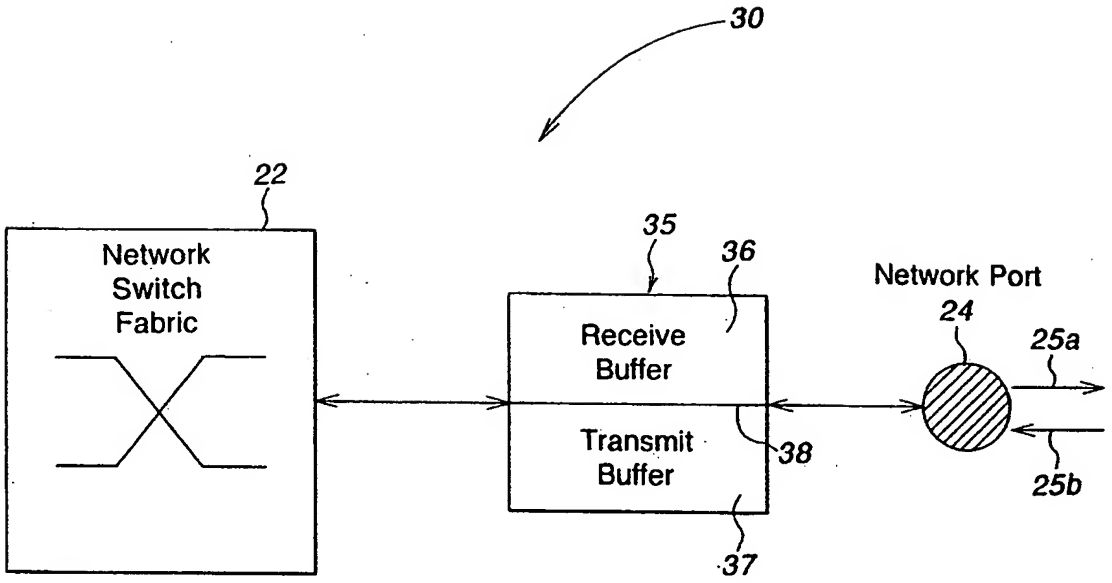


FIG. 3
(PRIOR ART)

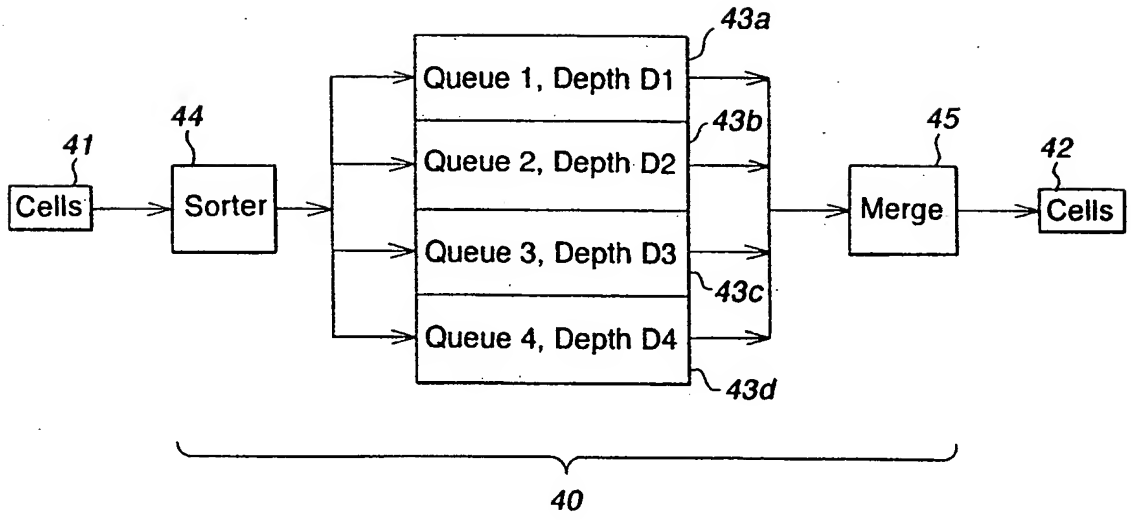

43a
Queue 1, Depth D1
43b
Queue 2, Depth D2
Queue 3, Depth D3
43c
Queue 4, Depth D4
43d
41
Cells
44
Sorter
45
Merge
42
Cells
40

FIG. 4

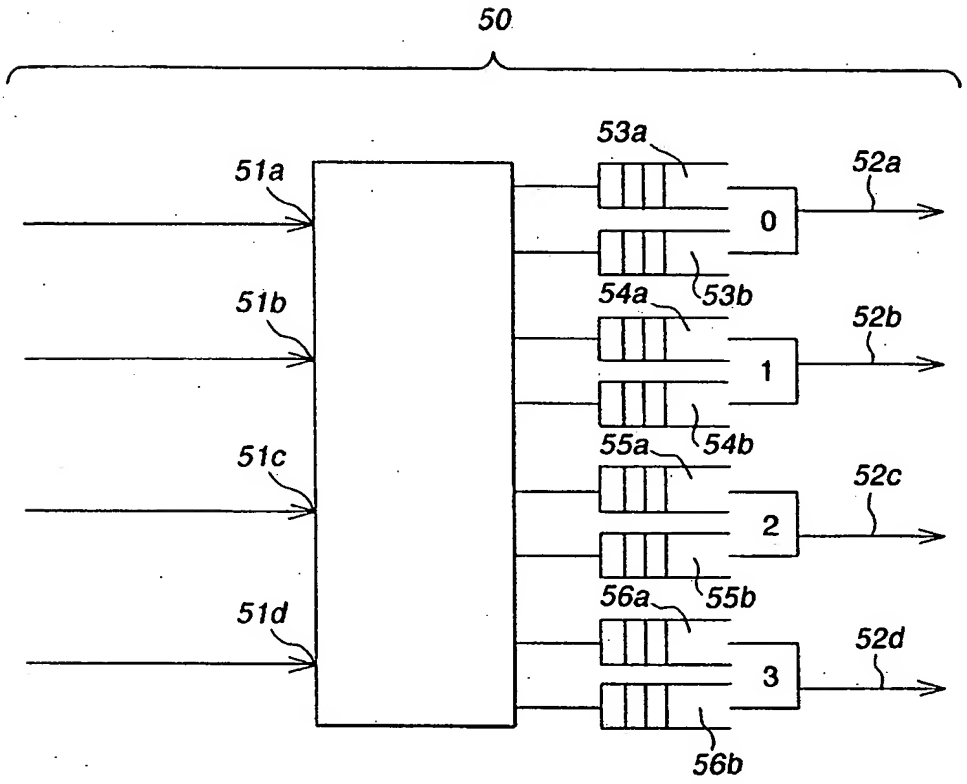

50
51a
51b
51c
51d
53a
53b
54a
54b
55a
55b
56a
56b
52a
52b
52c
52d
0
1
2
3

FIG. 5

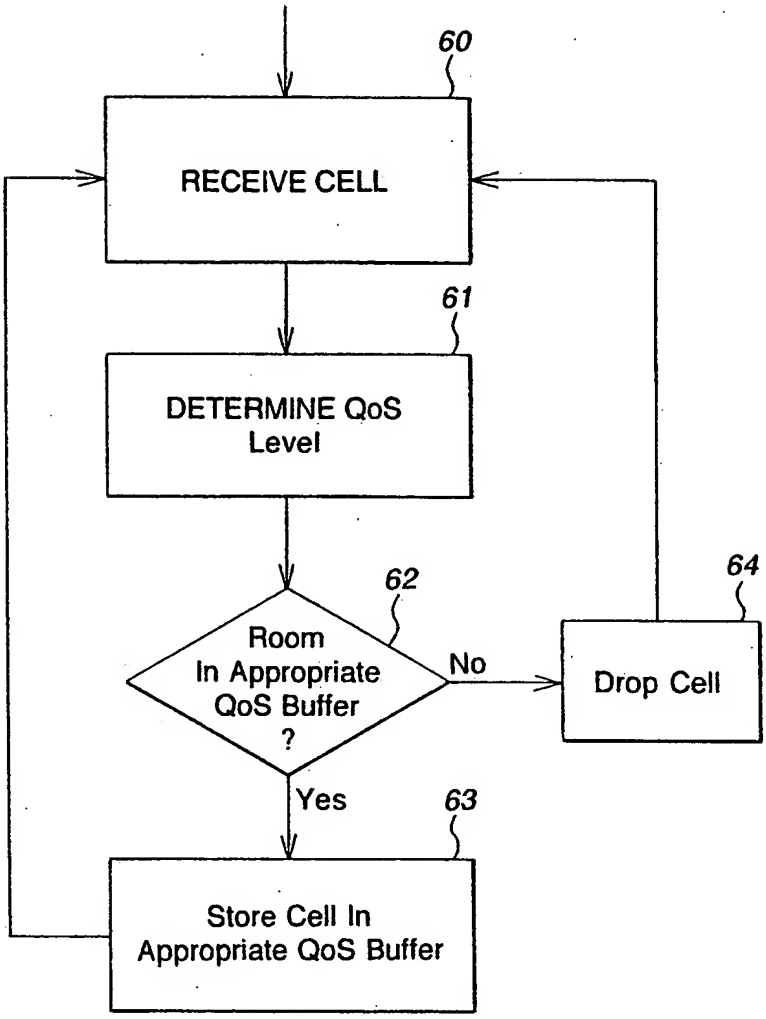

60
RECEIVE CELL
61
DETERMINE QoS Level
62
Room In Appropriate QoS Buffer ?
No
64
Drop Cell
Yes
63
Store Cell In Appropriate QoS Buffer

FIG. 6

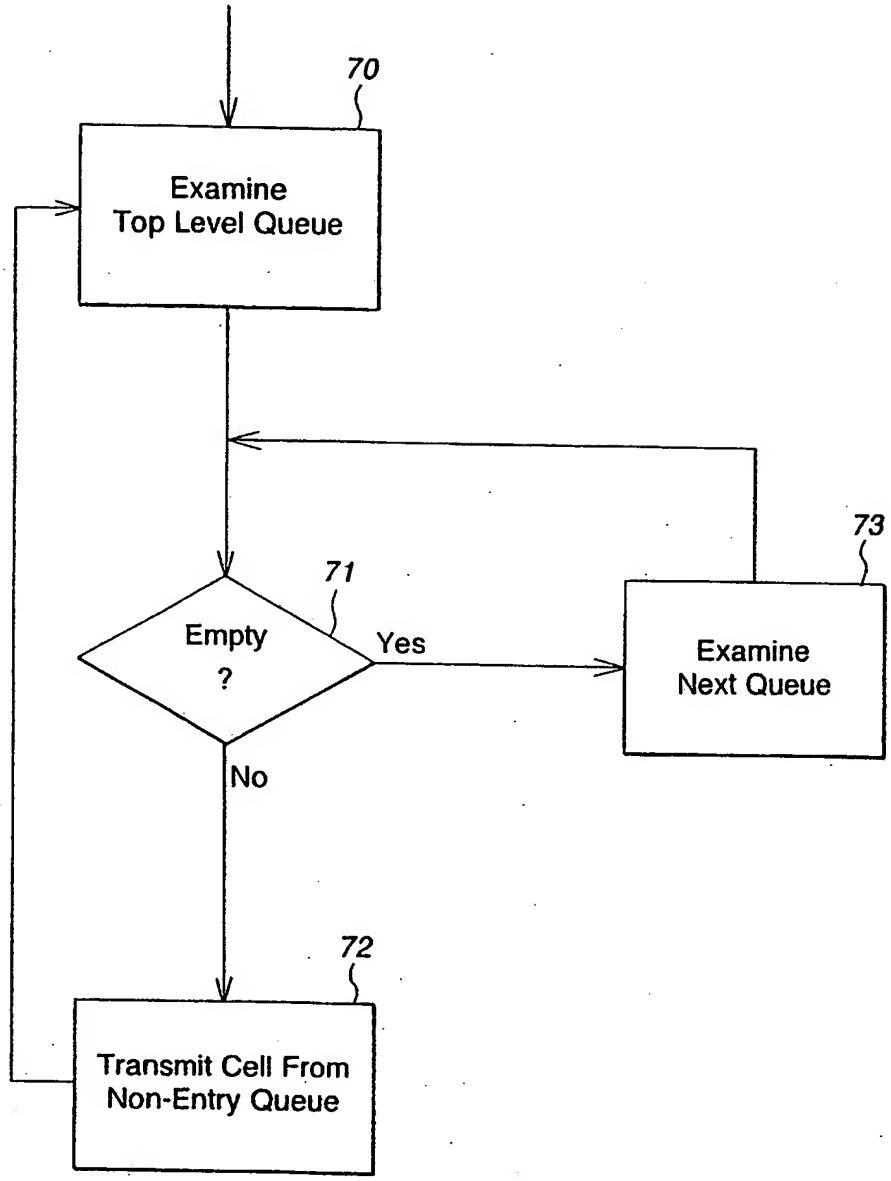

70
Examine
Top Level Queue
71
Empty
?
Yes
73
Examine
Next Queue
No
72
Transmit Cell From
Non-Entry Queue

FIG. 7

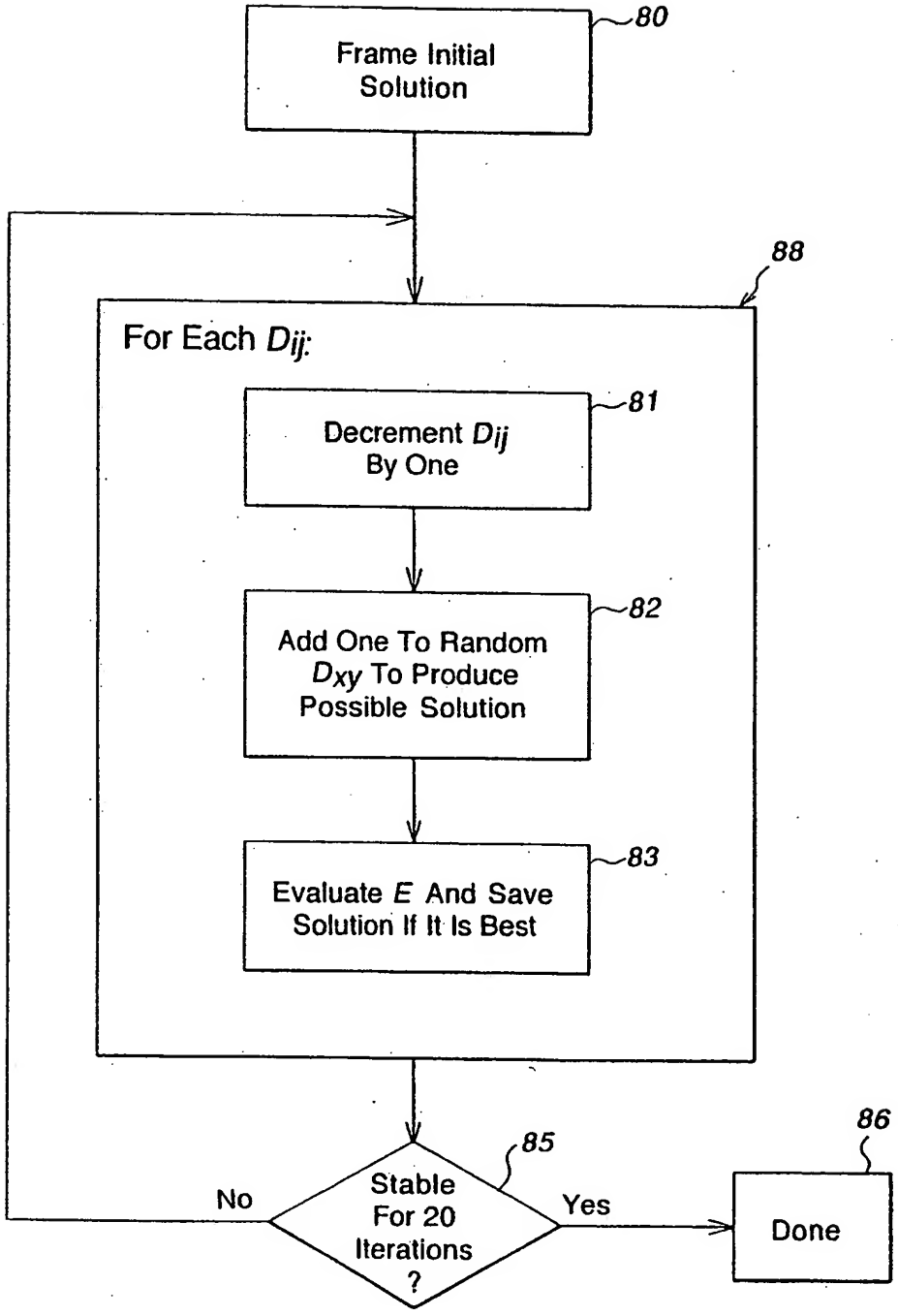

Frame Initial Solution
80
88
For Each Dij:
Decrement Dij By One
81
Add One To Random Dxy To Produce Possible Solution
82
Evaluate E And Save Solution If It Is Best
83
Stable For 20 Iterations ?
85
No
Yes
Done
86

FIG. 8

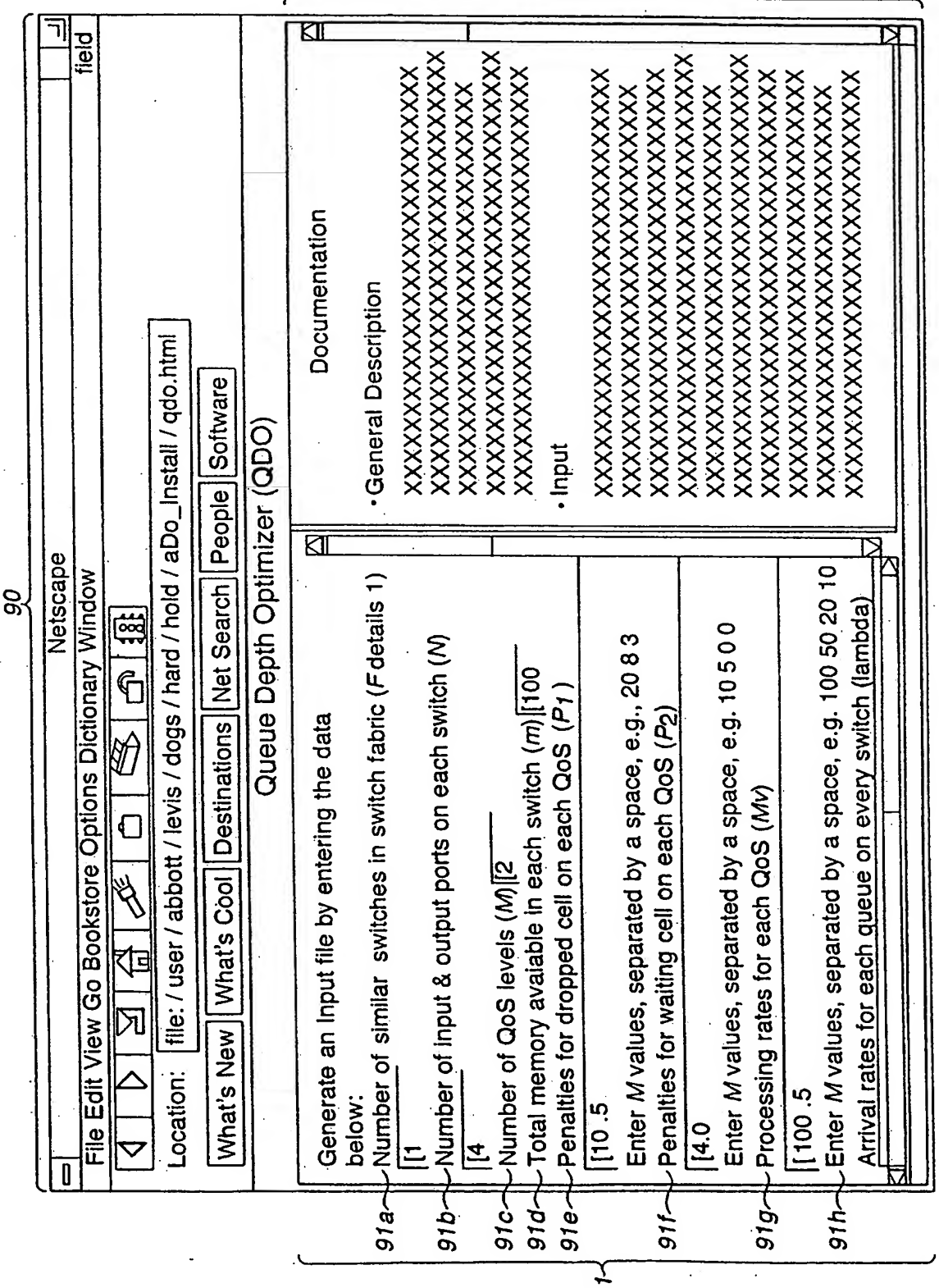

90
Netscape
File Edit View Go Bookstore Options Dictionary Window
field
Location: file: / user / abbott / levis / dogs / hard / hold / aDo_Install / qdo.html
What's New
What's Cool
Destinations
Net Search
People
Software
Queue Depth Optimizer (QDO)
Generate an Input file by entering the data below:
91a Number of similar switches in switch fabric (F details 1)
[1
91b Number of input & output ports on each switch (N)
[4
91c Number of QoS levels (M) [2
91d Total memory avaiable in each switch (m) [100
91e Penalties for dropped cell on each QoS (P1)
[10 .5
91f Enter M values, separated by a space, e.g., 20 8 3
Penalties for waiting cell on each QoS (P2)
[4.0
91g Enter M values, separated by a space, e.g. 10 5 0 0
Processing rates for each QoS (Mv)
[100 .5
91h Enter M values, separated by a space, e.g. 100 50 20 10
Arrival rates for each queue on every switch (lambda)
91
Documentation
92
• General Description
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
• Input
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

FIG. 9

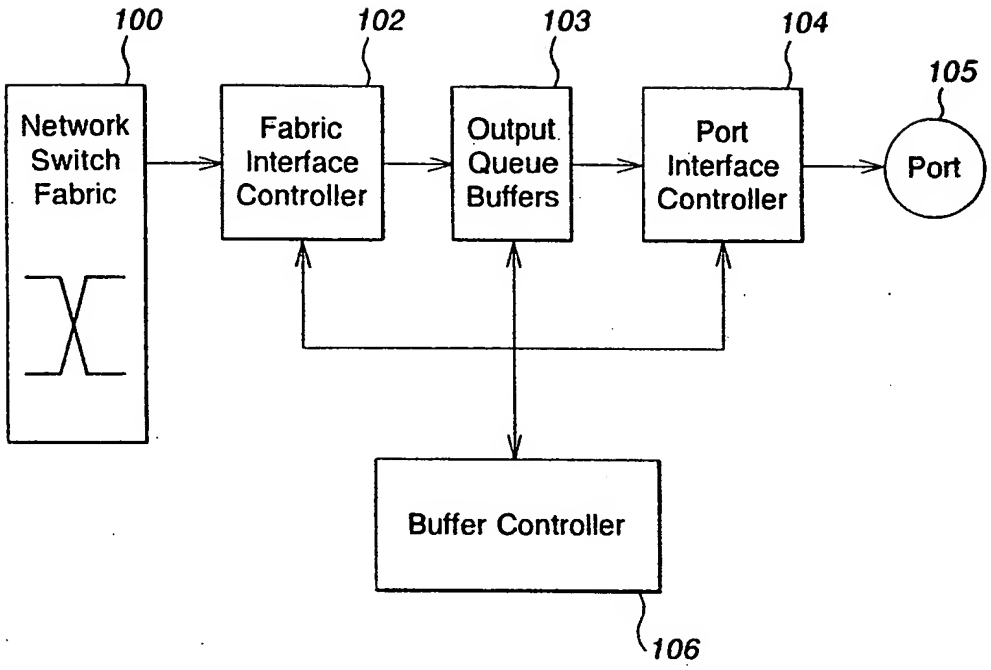

100
102
103
104
105
Network Switch Fabric
Fabric Interface Controller
Output Queue Buffers
Port Interface Controller
Port
Buffer Controller
106

FIG. 10

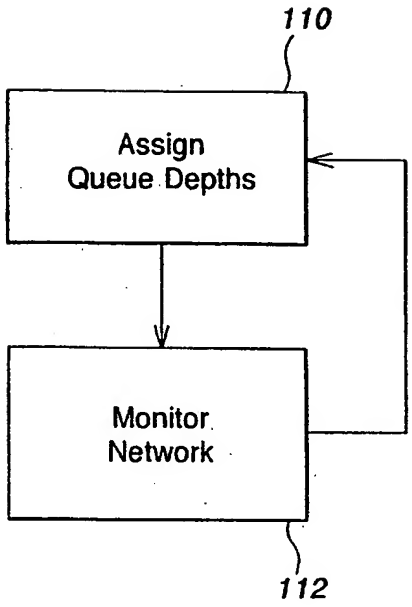


FIG. 11

INTERNATIONAL SEARCH REPORT

Int tional Application No
PCT/US 99/09853

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 555 265 A (KAKUMA SATOSHI ET AL) 10 September 1996 (1996-09-10) column 4, line 25 - column 5, line 40 column 7, line 50 - line 58 column 9, line 54 - line 62 column 10, line 34 - column 11, line 4 column 12, line 22 - line 36 column 13, line 28 - line 35 column 14, line 21 - column 15, line 65 --- -/--	1-32

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 August 1999	02/09/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Meurisse, W

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/09853

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 748 629 A (COLSMAN MATTHIAS L ET AL) 5 May 1998 (1998-05-05) column 2, line 16 - line 52 column 4, line 42 - line 44 column 5, line 30 - line 42 column 8, line 17 - column 9, line 12 column 11, line 7 - line 13 column 12, line 16 - line 28 column 13, line 13 - line 23 column 13, line 53 - column 14, line 7	7,32
A	G. I. ROBERTSON, J. F. MILLER AND P. THOMPSON: "Non-exhaustive methods and their use in the minimization of Reed-Muller canonical expansions" INTERNATION JOURNAL OF ELECTRONICS, vol. 80, no. 1, January 1996 (1996-01), pages 1-12, XP002112504 page 3, line 8 - line 41	4,5,27, 28

2

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int tional Application No
PCT/US 99/09853

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5555265 A	10-09-1996	JP 7240752 A	12-09-1995
US 5748629 A	05-05-1998	AU 6500796 A	18-02-1997
		AU 6500896 A	18-02-1997
		AU 6500996 A	18-02-1997
		AU 6501096 A	18-02-1997
		AU 6501496 A	18-02-1997
		AU 6501696 A	18-02-1997
		AU 6501796 A	18-02-1997
		AU 6501996 A	18-02-1997
		AU 6502096 A	18-02-1997
		AU 6502496 A	18-02-1997
		AU 6502596 A	18-02-1997
		AU 6502696 A	18-02-1997
		AU 6502796 A	18-02-1997
		AU 6503196 A	18-02-1997
		AU 6503296 A	18-02-1997
		AU 6503396 A	18-02-1997
		AU 6503496 A	18-02-1997
		AU 6503596 A	18-02-1997
		AU 6503696 A	18-02-1997
		AU 6503796 A	18-02-1997
		AU 6549196 A	18-02-1997
		AU 6549296 A	18-02-1997
		AU 6648496 A	18-02-1997
		AU 6648796 A	18-02-1997
		AU 6712496 A	18-02-1997
		AU 6712596 A	18-02-1997
		AU 6761896 A	18-02-1997
		AU 6762096 A	18-02-1997
		EP 0845181 A	03-06-1998
		EP 0839420 A	06-05-1998
		EP 0839419 A	06-05-1998
		EP 0872086 A	21-10-1998
		EP 0839421 A	06-05-1998
		EP 0839422 A	06-05-1998
		WO 9704558 A	06-02-1997
		WO 9704559 A	06-02-1997
		WO 9704560 A	06-02-1997
		WO 9704541 A	06-02-1997
		WO 9704561 A	06-02-1997
		WO 9703549 A	06-02-1997
		WO 9704562 A	06-02-1997
		WO 9704542 A	06-02-1997
		WO 9704552 A	06-02-1997
		WO 9704554 A	06-02-1997
		WO 9704544 A	06-02-1997
		WO 9704556 A	06-02-1997
		WO 9704563 A	06-02-1997
		WO 9704557 A	06-02-1997
		WO 9704543 A	06-02-1997